

**ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ**



**ИНФОРМАЦИОННЫЕ УПРАВЛЯЮЩИЕ СИСТЕМЫ И
КОМПЬЮТЕРНЫЙ МОНИТОРИНГ(ИУС КМ - 2015)**

**Сборник материалов VI Международной научно - технической
конференции студентов,
аспирантов и молодых ученых**

20-22 мая 2015 г.

Донецк, ДонНТУ – 2015

УДК 004

Информационные управляющие системы и компьютерный мониторинг в рамках форума «Инновационные перспективы Донбасса» (ИУС КМ - 2015): VI Международная научно-техническая конференция студентов, аспирантов и молодых ученых, 20-22 мая 2015, г.Донецк: / Донец. национал. техн. ун-т; сост. :К.Н.Маренич (председатель) и др. - Донецк: ДонНТУ, 2015. - 250 с.

В сборнике опубликованы результаты научных исследований и технических разработок в области современных информационных технологий, компьютерного мониторинга, искусственного интеллекта, моделирования, разработок цифровых устройств, экспертных систем диагностики, использования методов интеллектуального анализа данных, Web-технологий.

У збірнику опубліковані результати наукових досліджень та технічних розробок у сфері сучасних інформаційних технологій, комп'ютерного моніторингу, штучного інтелекту, моделювання, розробки цифрових пристроїв, експертних систем діагностики, використання методів інтелектуального аналізу даних, Web-технологій.

Results of scientific research and development works are published in collected papers in following fields: modern information technologies, computer monitoring, artificial intelligence, simulation, digital device development, diagnostic expert systems, usage of intelligent data analysis methods, Web-technologies.

Редакционная коллегия:

Маренич К. Н. - д.т.н., проф., проректор по научной работе ДонНТУ (председатель коллегии); Струнилин В. Н. - декан факультета КТН ДонНТУ; Скобцов Ю.А. - д.т.н., профессор, зав. кафедрой АСУ ДонНТУ; Аверин Г.В. - д.т.н., профессор, зав. кафедрой КСМ ДонНТУ; Новиков Д. Д. - аспирант кафедры АСУ ДонНТУ; Григорьев А. В. - к.т.н., доцент кафедры ПМИ, зам. декана по научной работе факультета КТН ДонНТУ; Светличная В.А - к.т.н., доцент кафедры АСУ ДонНТУ; Андриевская Н.К. - ст. пр. каф. АСУ; Секирин А.И. - доцент кафедры АСУ; Савкова Е. О., - доцент кафедры АСУ; Васяева Т.А.- доцент кафедры АСУ; Орлов Ю.К. - доцент кафедры САУ; Землянская С.Ю. - доцент кафедры АСУ; Копытова О.М. - доцент кафедры ПОиС; Павлий В.А. доцент кафедры КСМ; Дяченко О.Н. - доцент кафедры КИ.

Адрес редакционной коллегии

г.Донецк, ул. Артема 58, учебный корпус 8, ауд. 601

Веб-адрес конференции: <http://iuskm.donntu.org>

E-mail адрес: iuskm@donntu.org

© Донецкий национальный технический университет, 2015

СОДЕРЖАНИЕ

ПЛЕНАРНЫЕ ДОКЛАДЫ	
Григорьев А.В. Методы исредства автоматизации построения интеллектуальных САПР.....	
Андрюхин А.И., Подтынный С.Д. Вероятностный анализ индуктивных и рефлексивных утверждений.....	
Зинченко Ю.Е., Дяченко О.Н., Калашников В.И. FPGA-технологии проектирования и диагностика радиоэлектронной и электрлонно-вычислительной аппаратуры в ДонНТУ.....	
Бельков Д.В., Едемская Е.Н. Исследование UPD-трафика с помощью функции Харстра.....	
Харитонов А.В. Автоматизированная система учета коммунальной собственности г.Донецка как часть инновационного развития региона в рамках работы кафедры компьютерных систем мониторинга.....	
РАЗДЕЛ 1.ИНФОРМАЦИОННЫЕ УПРАВЛЯЮЩИЕ СИСТЕМЫ И ТЕХНОЛОГИИ	7
Иванова И. И., Землянская С. Ю. Исследование методов распознавания образов на снимке микрокапель в покрытии, нанесенного методом вакуумного дугового напыления.....	8
Крайний Д. С. Землянская С.Ю. Разработка система поддержки принятия решений при составлении спортивного календаря спортсменов-велосипедистов.	15
Матях И.В., Савкова Е. О. Разработка алгоритма для системы оценки социально-экономических показателей развития предприятия.....	22
Мартыненко Т.В., Терешкин Ю.С., Андриевская Н.К. Предобработка и оконтуривание рентгенографических изображений верхних конечностей.	28
Поляков А.И., Новиков Д.Д., Щербаков А.А. Анализ видов определения численности групп наблюдения для статистических расчетов показателей оценки эффективности лечения больных.....	38
Сафонов М.Д., Светличная В.А. Использование методов	47

поиска оптимального пути при построении маршрута для системы мониторинга водной среды.	
Чепикова Е.Д., Савкова Е.О. Исследование методов оптимизации структуры и контента сайта на основании данных аналитики.....	56
РАЗДЕЛ 2. КОМПЬЮТЕРНЫЙ МОНИТОРИНГ И МЕТОДЫ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ	63
Григорьев А.В., Заплетин Е.А. Основные подходы при создании рекомендательных систем.	64
РАЗДЕЛ 3. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, НЕЙРОСЕТЕВЫЕ И ЭВОЛЮЦИОННЫЕ АЛГОРИТМЫ, ЭКСПЕРТНЫЕ СИСТЕМЫ	71
В.В. Елифёров, О.И. Федяев. Формирование обучающего множества зависимости знаний студентов от их менталитета для двухслойных нейронных сетей.	72
РАЗДЕЛ 4. WEB-ТЕХНОЛОГИИ, ИНТЕРНЕТ И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ	77
Калинина Е.С., Чернышова А.В. Анализ протокола безопасности Kerberos.....	78
Клокова М.В., Чернышова А.В. Анализ средств защиты информации в корпоративных сетях.....	85
И.А. Офицеров, К.А. Пшеничный. Исследование применимости метода диаграмм последовательностей для защиты от угроз информационной безопасности при передаче данных по оптоволоконным каналам.....	92
Сипаков Д.С., Губенко Н.Е. Сравнительный анализ методов лингвистической стеганографии.....	99
Тимонин Р.С., Пшеничный К.А. Исследование применимости метода диаграмм деятельности в сфере информационной безопасности.	105
РАЗДЕЛ 5. ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ	119
Алёхина А.Г., Краснокутский В.А., Кривошеев С.В., Фильтрация шумов во встраиваемых системах	120
Варламова М. А. Применение вейвлет-преобразований	

в системах	124
Дяченко В.О. Дяченко, О.Н. Особенности применения двойственных полиномов для аппаратной реализации циклических кодов	130
Лашенко К.С., Соколов Н.Г., Гудаев О.А. Онтологическая лингвистика интернета.....	137
Лунтовская А.А. Процедурная генерация облаков и рендеринг неба в реальном времени с использованием графического процессора.....	146
Плитка А.Ю., Потопахин А.А., Гудаев О.А. Пятое измерение – программирование.....	152
Сверчков В. Л., Завадский Д. А., Кривошеев С. В., Завадская Т. В. Организация симулятора-тренажёра инженерной техники с использованием технологии распределенных вычислений.	158
Сипаков Д.С., Харитонов А.Ю. Сравнительный анализ времени исполнения математического решения на ПЭВМ и контроллере с FPGA-процессором.	164
Шеремет Н.Н. Исследование метода активных контуров и его модификаций.	172
РАЗДЕЛ 6. КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ	179
Гимадеев К.Д., Теплинский С.В., Франжев К.Е. Анализ и исследование таймеров WINDOWS и C#.	180
Мохначёв В.В., Елагин И.А., Теплинский С.В. Описание и обработка системных и пользовательских ошибок.....	186
Д.О. Макагонов, Н.А. Романовский, Д.М. Бочаров. Разработка и поддержка программного модуля системы управления проектами и задачами.	196
РАЗДЕЛ 7. МОДЕЛИРОВАНИЕ И СОВРЕМЕННЫЕ КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ	203
Миненко А.С. Математическое моделирование одного класса нелинейных систем с применением нечеткой логики.	204
Подлесный А.И., Григорьев А.В. Применение	

генетического алгоритма в системах автоматического проектирования.	210
Сандул Ю. О. Модель исполнителей службы технического обслуживания и ремонта оборудования.	213
РАЗДЕЛ 8. МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ.	221
Гавага И.С., Тарасова И.А. Разработка программного обеспечения системы управления патрульно-постовой службой милиции.	222
РАЗДЕЛ 9. СИСТЕМНЫЙ АНАЛИЗ И УПРАВЛЕНИЕ ОРГАНИЗАЦИОННЫМИ И ТЕХНИЧЕСКИМИ ОБЪЕКТАМИ	229
Бровкина А.С., Орлов Ю.К. Синтез информационно-справочной системы диагностирования и лечения болезней для медицинских заведений.	230
Ванжа Т.В., Орлов Ю.К. Разработка системы управления горным предприятием в условиях рисков.	236
Радевич К.В., Орлов Ю.К. Розробка системи мережевого планування при будівництві автодоріг і мостів.	245



**ИНФОРМАЦИОННЫЕ
УПРАВЛЯЮЩИЕ СИСТЕМЫ
И ТЕХНОЛОГИИ**

СЕКЦИЯ 1

УМК 004

И. И. Иванова, С. Ю. Землянская

Донецкий национальный технический университет
кафедра автоматизированных систем управления

E-mail: zsa@yale.ru

ИНФОРМАЦИОННАЯ СИСТЕМА ИДЕНТИФИКАЦИИ МИКРОКАПЕЛЬ В ПОКРЫТИИ, НАНЕСЕННОГО МЕТОДОМ ВАКУУМНОГО ДУГОВОГО НАПЫЛЕНИЯ

Аннотация

Иванова И. И., Землянская С. Ю. Исследование методов распознавания образов на снимке микрокапель в покрытии, нанесенного методом вакуумного дугового напыления. Выполнен анализ методов распознавания образов на снимке. Была выбрана предварительная последовательность действий для распознавания образов на изображении. Произведено сравнение результатов методов, на основе которых было принято решение какой из методов лучше использовать.

***Ключевые слова:** методы распознавания изображений, бинаризация, выделение контуров, фильтрация.*

Постановка проблемы. В современном мире информационных систем значительное место занимают системы, основной функциональностью которых является процесс идентификации веществ на снимке, или распознавание образов на снимке. К таким и относится информационная система идентификации микрокапель в покрытии для облегчения процесса аналитической обработки снимков для научных сотрудников лабораторий, специализирующихся на ионно-плазменных технологиях, к примеру, НОЦ «Ионно-плазменные технологии» МГТУ им. Н.Э. Баумана.

Актуальность разработки такого типа системы заключается в том, на текущий момент мало информационных систем, которые смогли бы с максимальной точностью определить на снимке из-под микроскопа микрокапли покрытия, нанесенного методом

вакуумного дугового напыления. Хотя стоит учесть тот факт, что уже существуют программные средства для идентификации порошковых изделий «CINDERELLA» и для металлических изделий - ВТ – 1600, но эти системы не учитывают особенности вещества в покрытии, проекция которого будет подвергаться исследованию.

Учитывая вышеуказанное, разработка пользовательского программного приложения будет весьма эффективным и рациональным решением для будущих потребителей системы.

Основные функциональные возможности проектируемой ИС:

- бинаризация входящего RGB изображения;
- выделение контуров микрокапель
- фаза логической обработки результатов фильтрации;
- определения физических характеристик микрокапли (радиус, площадь).

Анализ литературы. Описание методов выделения контуров, а также их эффективность, приведено в докладе Т.Т.Ч. Буя, В.Г. Спицына «Анализ методов выделения краев на цифровых изображениях» [1].

Цель статьи – ознакомиться с методами выделения контуров, а также выбрать наиболее эффективный из них, описать алгоритм для идентификации капли на изображении, а также для определения физических параметров капли.

Постановка задачи исследования. Имеется входящее изображение, приведенное на рис. 1, которое будет подвержено обработке, размером M на N пикселей, где M – количество пикселей вдоль оси Ox , а N – Oy . Каждый пиксель P_{ij} обладает рядом следующих свойств:

- R_{ij} – значение интенсивности красного P_{ij} пикселя;
- G_{ij} – значение интенсивности зеленого P_{ij} пикселя;
- B_{ij} – значение интенсивности черного P_{ij} пикселя;
- $IfObject$ - принадлежит ли P_{ij} пиксель объекту;
- $Border_{ij}$ – является ли P_{ij} пиксель границей объекта;
- $Center_{ij}$ – является ли P_{ij} пиксель центром капли.

Формально изображение можно представить следующим образом:

$$\text{Image} = \begin{bmatrix} P_{11} & \dots & P_{1N} \\ \dots & \dots & \dots \\ P_{M1} & \dots & P_{MN} \end{bmatrix}$$

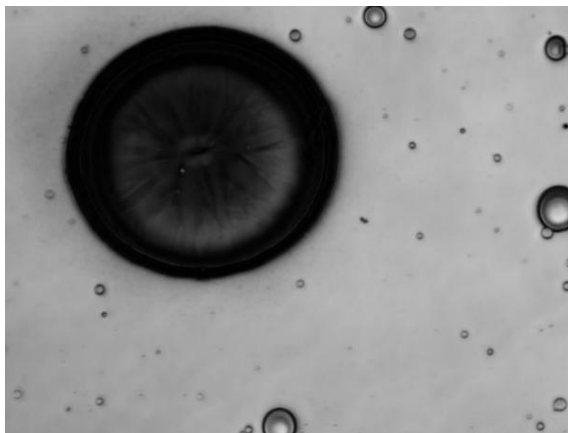


Рисунок 1 – Проекция покрытия в двумерном пространстве

Рассмотрение методов бинаризации изображения. Будет приведено описание метода Оцу.

Метод Оцу (англ. Otsu's method) — это алгоритм вычисления порога бинаризации для полутонового изображения, используемый в области компьютерного распознавания образов и обработки изображений.

Алгоритм позволяет разделить пиксели двух классов («полезные» и «фоновые»), рассчитывая такой порог, чтобы внутриклассовая дисперсия была минимальной. Метод Оцу также имеет улучшенную версию для поддержки нескольких уровней изображения, который получил название мульти-Оцу метод [2].

Метод Оцу ищет порог, уменьшающий дисперсию внутри класса, которая определяется как взвешенная сумма дисперсий двух классов.

В своей работе Оцу показывает, что минимизация внутриклассовой дисперсии эквивалента максимизации межклассовой дисперсии, которая равна:

$$\sigma_b^2 = w_1 w_2 (a_1 - a_2)^2$$

В этой формуле a_1 и a_2 — средние арифметические значения для каждого из классов. Особенность этой формулы заключается в том, что $w_1(t+1)$, $w_2(t+1)$, $a_1(t+1)$, $a_2(t+1)$ легко выражаются через предыдущие значения $w_1(t)$, $w_2(t)$, $a_1(t)$, $a_2(t)$ (t — текущий порог). Эта особенность позволила разработать быстрый алгоритм:

1. Вычисляем гистограмму (один проход через массив пикселей). Далее нужна только гистограмма; проходов по всему изображению больше не требуется.

2. Начиная с порога $t = 1$, проходим через всю гистограмму, на каждом шаге пересчитывая дисперсию $\sigma b(t)$. Если на каком-то из шагов дисперсия оказалась больше максимума, то обновляем дисперсию и $T = t$.

3. Искомый порог равен T . [3]

Описание рассмотренных методов для выделения границ.

Для выделения краев на цифровых изображениях могут быть использованы классические методы Собеля и Лапласа [4], а также выделение краев на основе применения математической морфологии [5].

Метод Собеля. Идея этого метода основана в наложении на каждую точку изображения двух масок вращения. Эти маски представляют собой две ортогональные матрицы размерностью 3×3 , вид которых приведен в таблице 1.

Таблица 1 «Маски вращения метода Собеля»

-1	0	+1	+1	+2	+1
-2	0	+2	0	0	0
-1	0	+1	-1	-2	-1

Эти маски выявляют границы, расположенные вертикально и горизонтально на изображении. При раздельном наложении этих масок на изображение можно получить оценку градиента по каждому из направлений G_x , G_y . Конечное значение градиента определяется по формуле: $G = \sqrt{G_x^2 + G_y^2}$

В методе Лапласа осуществляется умножение каждого элемента двумерной апертуры 3×3 на соответствующий элемент так называемой матрицы Лапласа, которая приведена на рис. 2.

1	1	1		0	-1	0		-1	1	1		1	1	1
1	-2	1		-1	4	-1		-1	-2	1		-1	-2	1
-1	-1	-1		0	-1	0		-1	1	1		-1	-1	1

Рисунок 2 – Матрицы Лапласа

Математическая морфология предназначена для исследования структуры некоторых множеств однотипных объектов. Любое изображение в компьютерной графике также обычно представляется в виде набора пикселей, поэтому операции математической морфологии могут быть применены и к изображению – для исследования некоторых свойств его формы и структуры, а также для его обработки [5].

Сравнение эффективности алгоритмов методов выделения границ. В своем докладе Т.Т.Ч. Буй также приводит результаты выполнения алгоритмов. Им было написано приложения средствами C#, реализующего методы выделения краев. Это приложение обрабатывает файлы формата *.gif, размером 256x256 пикселей. На рис. 4, 5 представлены результаты численных экспериментов по обработке изображений, а в таблице 1 – их сравнение [1].

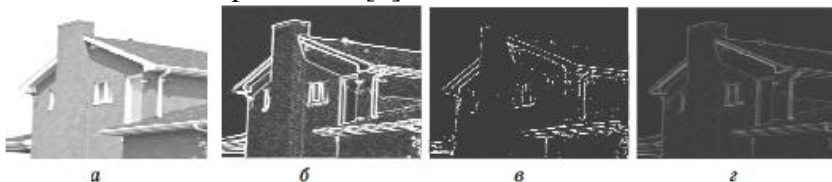


Рисунок 3 - Исходное изображение «House» – а; б – результат выделения краев методом Собеля; в – результат выделения краев методом Лапласа; г – результат выделения краев методом математической морфологии



Рисунок 4 - Исходное изображение «CameraMan» – а; б – результат выделения краев методом Собеля; в – результат

выделения краев методом Лапласа; γ – результат выделения краев методом математической морфологии

Таблица 2 «Сравнение времени выполнения алгоритмов»

Исходное изображение	Время выполнения, с		
	Собеля	Лапласа	Математическая морфология
«House»	2,4	2,57	1,12
«CameraMan»	1,77	1,9	0,84

Сравнение результатов обработки изображений указанными алгоритмами показало, что метод на основе применения математической морфологии работает быстрее, чем остальные методы. Следует отметить, что наиболее четкие границы объектов на изображении получаются при обработке на основе применения оператора Собеля.

Решение задач и результаты исследований. Для решения поставленной задачи предлагается исследовать метод для выделения границ на основе оператора Собеля, бинаризация изображения. Выбранная последовательность действий приведена на рис. 5.



Рисунок 5 – Последовательность действий при идентификации капли

Критерием того, что распознанный объект – капля, является наличие блика в центре окружности. Под бликом понимается группа пикселей по вертикали и горизонтали, у которых значения RGB резко отличаются от пикселей соседей. На бинарном изображении эта группа из последовательно расположенных пикселей со значениями $R=0$, $G=0$, $B=0$.

Выводы. Были изучены методы выделения границ микрокапель на снимке, а также было установлено какой из рассмотренных методов целесообразней использовать для задачи и соответственно был сделан выбор в пользу лучшего из них.

ы

Список литературы

1. Т.Ч. Буй, В.Г. Спицын. «Анализ методов выделения краев на цифровых изображениях»
2. 2.Материал из Википедии о методе Оцу
3. https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%9E%D1%86%D1%83
4. Обнаружение объектов методом Оцу
<http://habrahabr.ru/post/112079/>
5. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2005. – 1070 с.
6. 5. 3. Huang C.P. An Integrated Edge Detection Method Using Mathematical Morphology / C.P. Huang, R.Z. Wang // Pattern Recognition and Image Analysis. – 2006. – Vol. 16, № 3. – P. 406–412.

УМК 004

Д.С. Крайний, С.Ю. Землянская
Донецкий национальный технический университет
кафедра автоматизированных систем управления
E-mail: zsa@yall.ru

РАЗРАБОТКА СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ СОСТАВЛЕНИИ СПОРТИВНОГО КАЛЕНДАРЯ СПОРТСМЕНОВ-ВЕЛОСИПЕДИСТОВ

Аннотация

Крайний Д. С., Землянская С.Ю. Разработка система поддержки принятия решений при составлении спортивного календаря спортсменов-велосипедистов. Сформулирована цель разработки системы, рассмотрены существующие информационные системы, выделена функциональность разрабатываемой системы.

Ключевые слова: информационная система, СППР, велоспорт, рекомендации

Постановка задачи.

На данный момент процесс учета физического состояния профессионального спортсмена является обязательным фактором для достижения максимального результата. Это связано с тем, что профессиональный велоспорт характеризуется высокой конкурентоспособностью между спортсменами, что накладывает высокие требования на уровень физической подготовки.

Актуальность разработки такого типа системы заключается в том, что благодаря развитию компьютерных технологий все данные, характеризующие тренировку и физическое состояние спортсмена, находятся в электронном виде. Это в свою очередь позволяет легко манипулировать ими, сводить данные в таблицы и графики, а также производить расчеты для составления рекомендации спортсмену.

Учитывая вышеуказанное, разработка системы поддержки принятия решений при составлении спортивного календаря

спортсменов-велосипедистов является весьма эффективным и рациональным решением для будущих потребителей системы.

Цель статьи.

Ознакомится с существующим процессом работы информационной системы, выявить возможные недостатки. Определить функциональность разрабатываемой информационной системы, выбрать методы решения поставленных задач.

Цель разработки системы.

Разрабатываемая система несет учетный характер и ориентирована на профессиональных спортсменов велосипедистов, а также спортсменов любителей, имеющих необходимые устройства для считывания их результатов, которые послужат входными данными для разрабатываемой системы.

Назначение системы оценить физическую подготовку спортсмена на данный момент, на основании временной статистики его тренировок. Эта информация может использоваться как для составления спортивного календаря на год, так и для распределения ролей гонщиков в команде. Система должна дать эти рекомендации, тем самым выполнить работу спортивного аналитика и помочь спортивному директору принять решение о составе команд на различные гонки в текущем сезоне.

Цель системы помочь выставить максимально эффективный состав спортсменов в их лучшей форме на определённую гонку, чтобы команда смогла показать лучший результат, что вероятно приведет к новым вложениям спонсоров.

Описание структуры и процесса функционирования системы.

Входные параметры загружаются в систему с велокомпьютера, подключенного к персональному компьютеру. К этим данным относятся пульс, скорость, каденс, мощность, температура воздуха, высота. Эти показатели представляются в виде массивов текущих значений на каждый момент времени замеров датчиков. Также записывается длительность и продолжительность поездки.

Первичная обработка данных заключается в том, что рассчитываются средние, максимальные и минимальные значения для каждого из входных параметров за всю дистанцию и небольшими интервалами, например, по одному километру.

После таких операций есть возможность отобразить статистические графики и таблицы.

На основании данных нескольких тренировок можно судить о прогрессе или регрессе спортсмена в плане физической формы. Точность результат будет зависеть от количества проведенных тренировок. Необходимо следить за изменениями в значениях параметров пульса, скорости, каденса и мощности при приблизительно одинаковых значениях температуры воздуха и набора/потери высоты. В случае, если похожих условий не найдено пользователю предлагается сравнить тренировку без учета факторов окружающей среды, сравнивая тренировку с результатами всех проведенных ранее тренировок. Если результаты спортсмена на протяжении нескольких тренировок ухудшаются, то можно судить о недостаточном отдыхе и необходимости снижения нагрузки.

Также на основании данных тренировок можно судить о наиболее подходящей специализации для спортсмена. Для этого определим возможные специализации: спринтер – человек с лучшим финишем, это можно определить по способности развить мощность более 700 Ватт, горняк – человек с лучшей средней скоростью при наборе высоты более 1000 м на 100 км пути, раздельщик – человек с лучшим индивидуальным ходом, это определяется относительно стабильной мощностью и высокой средней скоростью при наборе высоты менее 500 м на 100 км пути. Для корректного определения специализаций необходимо чтобы в системе присутствовали несколько гонщиков, а лучше вся команда.

Имея данные о подходящей специализации гонщика и календарный план гонок на предстоящий сезон можно составить гоночный индивидуальный календарь спортсмена. Спортивным директором каждой гонке выставляется приоритет от 0 до 10 и в зависимости от этого значения программа подбирает гонщиков на

гонку, чем выше приоритет, тем более сильный гонщик в необходимой категории будет выбран. Сила гонщика определяется условным коэффициентом, высчитываем на основании сравнения его средней скорости с другими гонщиками из его специализации. Каждая гонка требует гонщиков определенных специализаций, в отличие от туров, где есть гонки всех типов, а соответственно нужны гонщики всех классификаций.

Для корректного подбора спортсменов на предстоящую гонку необходимо иметь достаточно информации о самой гонке. В первую очередь различают однодневные и многодневные гонки. Для многодневной гонки важна такая особенность спортсмена, как восстанавливаемость. Это определяется по периодичности и результативности тренировок, например, если у гонщика есть более промежуток тренировок длительностью не менее 5-ти дней со средним пульсом более 150 ударов/мин – можно судить о хорошей восстанавливаемости спортсмена и имеет смысл рекомендовать его на многодневные гонки. Также отличительной чертой многодневных гонок является разнообразие этапов, что говорит о необходимости спортсменов различных категорий в составе команды.

Каждая гонка, а в случае тура – этап, должна быть описана в системе, и содержать информации о длительности, наборе высоты и спуске, максимальном градиенте, дате проведения и коротком описании.

Стоит отметить что программа поддерживает учет множества спортсменов, для этого используются учетные записи, хранящие информацию о пользователе - ФИО, дата рождения, рост, вес, контактный телефон.

Существует множество сторонних факторов, не поддающихся компьютерному анализу, поэтому выбранная классификация спортсмена и предложенный календарь гонок можно отнести к понятиям нечеткой логики.

На основании приведенной информации можно сказать что выходными параметрами будут гоночный календарь команды, а

также статистические таблицы и графики с показателями тренировок.

Недостатки существующей ИС.

На данный момент этот процесс частично автоматизирован. Система автоматически считывает данные с устройства, сохраняет их, отображает статистику в текстовом и графическом виде, но решение об изменении тренировок, о выборе гонок и роли спортсмена в гонке выбирает только спортивный директор и тренер. Главный недостаток такого подхода – длительность процесса анализа. Если автоматизировать данный процесс, выбор решения займет намного меньше времени, а также облегчит работу спортивного директора и тренера.

Выбор роли спортсмена в команде, а также изменения программы тренировок зависит от множества факторов, некоторые из которых довольно сложно предугадать, либо они зависят от действий спортсменов других команд. Поэтому к недостаткам разрабатываемой системы можно отнести возможную погрешность результатов работы, то есть выдаваемые рекомендации, именно поэтому конечное решение остается за спортивным директором и тренером, а система только предлагает им возможные варианты, значительно упрощая их работу.

Описание функциональности разрабатываемой ИС.

В плане функциональности можно выделить три основные функции системы:

1. Подбор классификации по результатам тренировок
2. Анализ предстоящих гонок
3. Подбор команды на гонку

Описание процесса выполнения функции «Подбор классификации по результатам тренировок»

Важным этапом данной функции является обработка входных значений и составление статистической информации, на основании которой генерируется вывод о наиболее подходящей квалификации для спортсмена. Статистические графики и таблицы, содержат средние, максимальные и минимальные значения с каждой тренировки по таким показателям как пульс,

скорость, каденс, мощность, сила ветра, набор/потеря высоты. На графике данные отображаются в виде кривых, построенных по точкам – по оси X – дата тренировки, по оси Y – значение параметра. Поскольку значения параметров будут сильно отличаться, например, отображение на одной шкале значения температуры воздуха и пульса как минимум будет выглядеть неясно, по оси Y график будет разделен на отдельные области. Пример приведен на рис. 1.

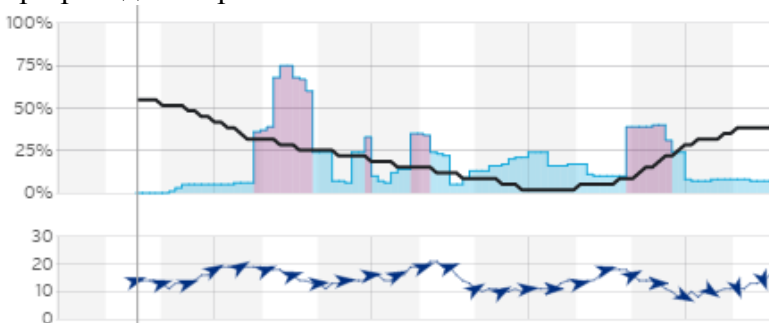


Рисунок 1 – Пример отображения статистического графика

Описание процесса выполнения функции «Анализ предстоящих гонок»

Для реализации подбора подходящей гонки для спортсмена необходимо иметь информации о гонке. Поскольку подготовка к гонке начинается за длительный период до старта множество факторов учесть нельзя, например, такие характеристики окружающей среды как температурный режим, силу ветра, осадки и т.п. Поэтому в программе используется минимальный набор характеристик гонки, а именно – количество метров набора высоты и спуска, максимальный градиент, длительность, раздельный или групповой старт.

На основании этих данных можно судить о наиболее подходящей квалификации гонки. В случае набора высоты более 1500 м – горная гонка, длительность менее 70 км – одиночная или командная разделка, а в случае набора высоты менее, чем на горных этапах и последних 10 км гонки с набором высоты не более 200 м – спринтерская гонка.

Описание процесса выполнения функции «Подбор команды на гонку»

Данная функция ориентирована на помощь в процессе принятия решения тренером и спортивным директором о подборе состава команды спортсменов на гонку.

Процесс выполнения функции начинается с того, что пользователь выбирает гонку, для которой необходимо выбрать состав команды. Далее на основании данных о гонке и тренирующихся спортсменов, полученных из вышеописанных функций система подбирает наиболее подходящих кандидатов на данную гонку.

Планируется использовать нейронные сети для реализации данной задачи. Система должна на основании входных данных предлагать наиболее оптимальный состав гонщиков на выбранную гонку. В качестве входных данных будет использоваться информация о спортсмене и гонке, а то, насколько профиль гонщика подходит под конкретную гонку, за счет сравнения данных с тренировок гонщиков с похожей классификацией.

Выводы

В статье сформулирована задача разработки системы поддержки принятия решений при составлении спортивного календаря спортсменов-велосипедистов, рассмотрены основные недостатки существующей информационной системы, предложено их решение. Выделена функциональность разрабатываемой системы.

Список литературы

1. О.И.Ларичев. Теория и методы принятия решений. М., Логос, 2000.
2. Ф. Уоссермен. - Нейрокомпьютерная техника: Теория и практика, 1992
3. Шеннон Совндаль, Анатомия велосипедиста – Попурри, 2011.
4. Д.А. Полищук, «Велосипедный спорт», 1997

И. В. Матях, Е. О. Савкова

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления

РАЗРАБОТКА АЛГОРИТМА ДЛЯ СИСТЕМЫ ОЦЕНКИ СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ ПОКАЗАТЕЛЕЙ РАЗВИТИЯ ПРЕДПРИЯТИЯ

Аннотация

Матях И.В., Савкова Е. О. Разработка алгоритма для системы оценки социально-экономических показателей развития предприятия. Выполнен анализ оценки уровня социально-экономического развития предприятия. Выбраны основные группы показателей для оценки. Описан процесс определения системы показателей и оценки уровня развития предприятия.

***Ключевые слова:** оценка уровня социально-экономического развития предприятия, группы показателей, отраслевой признак, формы собственности, кластеризация.*

Постановка проблемы. Нестабильная ситуация и быстро изменяющаяся внешняя среда на сегодняшний день оказывает влияние на развитие предпринимательской деятельности. Для улучшения уровня социально-экономического развития предприятий, необходимо, в первую очередь выявить имеющиеся проблемы и провести их диагностику (анализ). В основном, главными проблемами на предприятии можно назвать следующие: низкое качество продукции и услуг, нерациональное использование ресурсов, неудовлетворение потребностей работников предприятия.

Актуальность разработки такого типа системы заключается в том, что на данный момент, не существует системы, используя которую можно оценить текущее состояние предприятия, основные проблемы (если они имеются) и их решения, а также получить предложения о дальнейшем его развитии.

Учитывая вышеуказанное, разработка такого алгоритма будет весьма эффективным решением для будущих потребителей системы.

Анализ литературы. Описание основных этапов оценки уровня социально-экономического развития предприятия, приведено в докладе Т.В. Калинеску, Е. А. Недобега «Анализ и оценка уровня социально-экономического развития предприятия» [1].

Цель статьи – ознакомиться с основными этапами анализа уровня развития предприятия, описать алгоритм выбора социально-экономических показателей развития предприятия.

Постановка задачи исследования. Проблема развития предприятия рассматривалась многими учеными. В работе [1] предлагается следующая обобщенная схема анализа и оценки уровня социально-экономического развития предприятий:

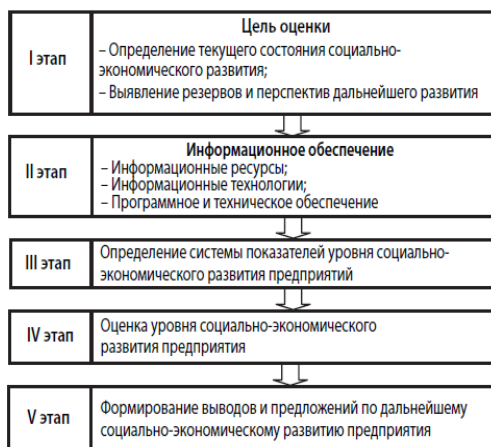


Рисунок 1 - Схема анализа и оценки уровня развития предприятий

Предложенная схема включает в себя несколько этапов. На первом этапе определяется цель проведения оценки. Далее проводится сбор всей необходимой информации, в зависимости от которой формируется система показателей. На двух последних этапах производится оценка уровня развития предприятия, и

формируются предложения по решению проблем и дальнейшему развитию предприятия.

Для более точной оценки уровня социально-экономического развития предприятия необходимо провести анализ его деятельности (классифицировать его по ряду признаков). Данный анализ влияет на выбор необходимых групп показателей для данного предприятия.

Предприятия различаются между собой большим количеством признаков. Самым главным считается отраслевой признак [2]. По отраслевому признаку предприятия подразделяются на:

- промышленные предприятия, которые осуществляют добычу и переработку полезных ископаемых, и производство товаров. Промышленность можно разделить на две большие отраслевые группы:

- добывающая и перерабатывающая промышленность. В свою очередь, перерабатывающая промышленность делится на отрасли легкой, пищевой, тяжелой промышленности и т. д.

- сельскохозяйственные предприятия, производящие продукцию растениеводства, животноводства и др.;

- торговые предприятия, которые сами не производят товары, но выполняют дистрибьюторские функции;

- предприятия строительства;

- банки, которые собирают финансовые средства и предоставляют кредиты;

- транспортные предприятия, которые занимаются перевозками с использованием различных транспортных средств;

- страховые организации, которые осуществляют страхование от различных видов рисков;

- предприятия в сфере услуг (например, гостиницы, консалтинговые фирмы и другие).

По формам собственности предприятия бывают:

- государственные;

- коллективные;

- смешанные;

- частные.

По сфере деятельности существуют такие предприятия:

- материального производства (это в основном промышленные и сельскохозяйственные, строительные предприятия и др.);
- нематериального производства, которые предоставляют услуги - от ремонта бытовой техники до услуг здравоохранения, образования, культуры;

По структуре производства предприятия делятся на:

- узкоспециализированные (выпускают ограниченный ассортимент продукции массового или крупносерийного производства, например, сталь);
- многопрофильные (выпускают продукцию широкого ассортимента или различного назначения);
- комбинированные (производства, в которых один вид продукции или сырья превращается в другой вид. Например, сырье - пряжа - ткань).

По мощности производственного потенциала предприятия делятся на три группы: малые, средние, крупные. В основу классификации заложены следующие признаки: численность персонала, стоимостной объем выпуска продукции, стоимость средств производства. Для группировки предприятий по мощности используются различные нормативы. Например, по численности выделяют малые предприятия: в промышленности до 100 человек, в сельском хозяйстве до 60 человек, в розничной торговле до 30 человек, в прочих предприятиях до 50 человек [3]. При проведении анализа предприятия, классификация его по вышеперечисленным признакам играет важную роль, так как каждый из данных признаков влияет на выбор показателей и, соответственно, на формирование оценки уровня развития предприятия. Если проводить классификацию по определенному признаку (например, по отраслевому), то это повлияет на формирование выводов: предприятие будет рассмотрено только с одной стороны и в этом случае, вывод будет содержать детальную оценку развития предприятия с учетом его отраслевой принадлежности. Такая оценка важна, если необходимо провести определенный анализ предприятия, то есть рассмотреть его без

учета некоторых признаков. Изучив предложенную схему анализа и оценки уровня социально-экономического развития предприятий в работе [1], можно сделать вывод, что она не включает в себя важный этап классификации предприятия, который влияет на выбор показателей и соответственно, на формирование выводов. В связи с этим, данная схема была модифицирована:



Рисунок 2 – Общая схема оценки уровня развития предприятий

Процесс определения системы показателей и оценки уровня развития предприятия. Систему показателей можно разделить на два основных раздела: экономические и социальные. В каждом разделе необходимо провести группировку (кластеризацию), объединяющую показатели [4]. Например, в группе «Охрана труда» можно выделить показатели расходов на охрану труда, уровень профессиональных заболеваний и т.д. Каждая группа имеет вес (значимость) и шкалу оценивания. После заполнения всех показателей в группе, формируется общее

значение (оценка) для данной группы, которая в дальнейшем нормализуется для получения общего вывода. Например, при оценивании показателей охраны труда, в зависимости от полученных результатов, можно определить, поддерживает ли предприятие на надлежащем уровне условия труда и жизни трудового коллектива. На данном этапе выделены следующие основные группы показателей, представленные в таблице 1.

Таблица 1 «Основные группы показателей»

Экономические показатели	Социальные показатели
Группа показателей эффективности работы предприятия	Группа показателей охраны труда
Группа показателей инновационной деятельности	Группа показателей организации труда
Группа показателей использования оборотных средств	Группа показателей образования работников
Группа показателей рентабельности	Группа экологических показателей

Выводы. Данная система играет важную роль для предпринимателей, которые хотят получить как комплексную оценку своего предприятия, так и оценку в зависимости от определенных факторов (направлений работы), определить проблемы и получить предложения по их решению и дальнейшему развитию предприятия.

Список литературы

1. Т.В. Калинеску, Е. А. Недобега «Анализ и оценка уровня социально-экономического развития предприятия».

2. Классификация предприятий/ Интернет-ресурс. - Режим доступа: www/ URL:

http://edu.dvgups.ru/METDOC/ITS/EKON_S/EKON_PREDPR/UM_K_DO/FRAME/2_5.htm - Загл. с экрана.

3. Горфинкель В. Я. Экономика предприятия / В.Я. Горфинкель, В.А. Швандар – М.: «Юнити-дана», 2007. – 670 с.

4. А. Котов, Н. Красильников «Кластеризация данных».

УДК 004.932.2+004.932.72'1

Т.В. Мартыненко, Ю.С. Терешкин, Н.К. Андриевская
Донецкий национальный технический университет
кафедра автоматизированных систем управления
E-mail: yres22@gmail.com

ПРЕДОБРАБОТКА И ОКОНТУРИВАНИЕ РЕНТГЕНОГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ ВЕРХНИХ КОНЕЧНОСТЕЙ

Мартыненко Т.В., Терешкин Ю.С., Андриевская Н.К.
Донецкий национальный технический университет
кафедра автоматизированных систем управления
E-mail: yres22@gmail.com

Аннотация:

Мартыненко Т.В., Терешкин Ю.С., Андриевская Н.К. Предобработка и оконтуривание рентгенографических изображений верхних конечностей. Рассмотрена обработка и классификация рентгеновских изображений по видам переломов. Исследованы такие основные методы оконтуривания объектов изображений как: метод Канни, метод Собеля, бинаризация по порогу яркости с использованием метода четырёхсвязности. На основе анализа полученных результатов выбран метод для улучшения качества рентгенографического изображения и определения границ объектов.

Общая постановка проблемы

Переломы различного вида и степени тяжести были и остаются частым явлением в жизни человека. Для успешного восстановления перелома необходимо незамедлительное вмешательство врача. Дабы избежать негативных травматических факторов для пациента, при восстановлении поврежденной кости, необходимо повысить точность планирования врачебных действий.

Несмотря на технический прогресс в сфере компьютерной диагностики, рентгенография была и остается основным способом диагностирования опорно-двигательной системы человека. Применение методов компьютерной обработки рентгеновских изображений, позволяет улучшить качество снимков, автоматизировать определение основных характеристик и видов перелома.

В зависимости от формы и направления различают такие виды переломов, рисунок 1.

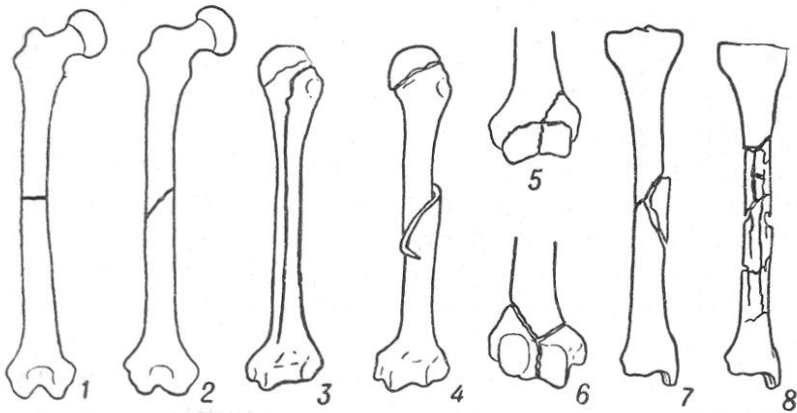


Рисунок -1.1 Схема переломов трубчатых костей: 1 — поперечный; 2 — косой; 3 — продольный; 4 — винтообразный; 5 — крестообразный; 6 — У-образный; 7 — клиновидный; 8 — оскольчатый.

1. Поперечные. Линия перелома условно перпендикулярна оси трубчатой кости.

2. Продольные. Линия перелома условно параллельна оси трубчатой кости.

3. Косые. Образуются при сгибании трубчатых костей. Линия перелома проходит под острым углом к оси трубчатой кости.

4. Винтообразные. Линия перелома идет по кости спирально .Происходит вращение костных отломков: костные отломки «повёрнуты» относительно своего нормального положения.

5. Оскольчатые. Кость в месте повреждения раздроблена на отдельные отломки.

6. Клиновидные. Одна кость вдавливается в другую, образуя клиновидную деформацию.

7. Вколоченные. Один из отломков внедряется в другой.

8.Отрывные. Возникают при внезапных сильных мышечных сокращениях при этом отрываются костные фрагменты, к



которым крепятся мышцы.

На рисунке 2 изображены примеры рентгенограмм костей человека.

Рисунок -.2 Пример рентгенограммы

Рассмотрев рентгенограмму можно сделать выводы о качестве изображения:

- изображение зашумлено;
- недостаточная контрастность и яркость;
- нечеткость границ объекта.

Таким образом, возникает необходимость спроектировать автоматизированную систему обработки рентгенографических изображений, которая обеспечит улучшение качества изображения и позволит определить вид перелома кости для дальнейших действий врача.

Для решения поставленной задачи необходимо:

- улучшить качество изображения: контраст, изменение яркости;
- провести бинаризацию изображения;
- выполнить оконтуривание объекта;
- выделить объекты на изображении;
- рассчитать характеристики объектов;
- определить вид перелома кости

При проектировании системы используется набор цифровых полутоновых([0..255]) рентгенографических изображений. Размер изображения находится в пределах от 100x100 до 450x500 px.

Основная задача статьи - определение метода выделения объекта на рентгенографическом изображении и его реализация для дальнейшей задачи определения вида перелома.

Предобработка рентгенографического изображения

До того как построить контур изображения, рентгенографическое изображение подвергается действиям по улучшению: балансируется яркость и контрастность, применяется фильтр уменьшающий шумы на изображении.

Яркость точки находится по формуле, коэффициенты которой определяются свойствами человеческого зрения:

$$Y = 0.3 * R + 0.59 * G + 0.11 * B$$

Изменение контрастности определяется по формуле:

$$NewY := K * (OldY - AveY) + AveY$$

NewY – новое значение одного из каналов, K – коэффициент контрастности (K=(0..1) – снижение, K<1 – повышение контрастности), OldY – текущее значение того же канала, AveY – среднее значение того же канала по изображению.

Определение контуров объекта на рентгенограмме

Некоторые задачи обработки и анализа изображений предусматривают выделение контура бинарного изображения для его дальнейшего использования в задачах распознавания и классификации. Для выделения контура изображения будут рассматриваться несколько способов: бинаризация по порогу яркости с использованием четырехсвязности, метод Канни, метод Собеля.

1) Выделение контура изображения с помощью пороговой бинаризации с применением метода четырёхсвязности.

Используем предложенный признак связности для вышеуказанной задачи [1, 2]. Пусть входное изображение представлено матрицей $B^0(M,N)$, элементы $b^0(m,n)$ которой принимают значения нуля (фон) или единицы (изображения).

1. Представим элементы бинарного изображения $B^0(M,N)$ значениями собственных связностей с соседними элементами, то есть вместо единичного элемента изображения записывается значения его связности. Будем называть такую модель изображения $Z(M,N)$ связностным изображением:

$$z(m, n) = \begin{cases} \delta^k(m, n), & \text{если } b^0(m, n) = 1 \\ 0 & \text{если } b^0(m, n) = 0 \end{cases}$$

2. Определяется стартовая точка на связностном изображении. Это может быть любая точка принадлежащая контуру. Она используется как первая ненулевая точка при последовательном просмотре изображения. Ей присваивается метка контура. То есть: если $z(m,n) \neq 0$, то $z(m,n)$ – стартовая и соответственно формируется матрица контурного изображения $B^1(M,N)$: $b^1(m,n)=1$.

3. Анализируются значения точек, которые граничат со стартовой. Среди них выбирается соседняя минимальная по значению ненулевая точка.

$$z_{min} = MIN\{z(m-1, n), z(m+1, n), z(m-1, n-1), z(m, n+1), z(m+1, n-1), z(m-1, n+1), z(m, n-1), z(m+1, n+1)\}$$

Ей также присваивается метка контура и осуществляется переход.

2) Метод Канни

Метод Канни является одним из лучших методов выделения контуров [3, 4]. Особенностью этого метода является использование морфологических операций. Алгоритм этого метода включает в себя следующие действия над изображением, из которого выделяются контуры:

- Сглаживание изображения посредством фильтрации, с использованием импульсной функции, описываемой гауссовым законом

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

где σ — параметр, определяющий степень сглаживающего действия импульсной функции на присутствующий в изображении шум. Чем больше σ , тем сильнее сглаживание.

- Нахождение градиента изменения яркости в изображении путем вычисления квадратного корня из суммы квадратов производных яркости изображения $L_c(x, y)$ по координатам x и y .

$$|G| = \sqrt{G_x^2 + G_y^2},$$

где $G_x = \frac{\partial L_c(x, y)}{\partial x}$, $G_y = \frac{\partial L_c(x, y)}{\partial y}$.

3) Метод Собеля

Оператор Собеля основан на свёртке изображения небольшими сепарабельными целочисленными фильтрами в вертикальном и горизонтальном направлениях, поэтому его относительно легко вычислять. С другой стороны, используемая им аппроксимация градиента достаточно грубая, особенно это сказывается на высокочастотных колебаниях изображения [5, 6].

Оператор вычисляет градиент яркости изображения в каждой точке. Так находится направление наибольшего увеличения яркости и величина её изменения в этом направлении. Результат показывает, насколько «резко» или «плавно» меняется яркость изображения в каждой точке, а значит, вероятность нахождения точки на грани, а также ориентацию границы. На практике, вычисление величины изменения яркости (вероятности принадлежности к грани) надежнее и проще в интерпретации, чем расчёт направления.

Оператор использует ядра 3×3 , с которыми сворачивают исходное изображение для вычисления приближенных значений производных по горизонтали и по вертикали. Пусть A исходное изображение, а G_x и G_y — два изображения, где каждая точка содержит приближенные производные по x и по y . Они вычисляются следующим образом:

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A \text{ and } G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A$$

где $*$ обозначает двумерную операцию свертки.

Координата x здесь возрастает «направо», а y — «вниз». В каждой точке изображения приближенное значение величины градиента можно вычислить, используя полученные приближенные значения производных:

$$G = \sqrt{G_x^2 + G_y^2}$$

Используя эту информацию, мы также можем вычислить направление градиента:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

где, к примеру, угол θ равен нулю для вертикальной границы, у которой тёмная сторона слева.

Результаты исследования способов оконтуривания

В результате исследования была разработана программа в программной среде Visual Studio 2010 на языке C#. Данная программа загружает выбранное изображение в пустую форму и при выборе одного из многих режимов выполняет определенную обработку над изображением. После выполненного действия изображение размещается в форму рядом с исходным. Экранная форма программы изображена на рисунке 3.



Рисунок -3.3 Экранная форма программы

На рисунке 4 показаны результаты получения контура объекта при применении пороговой бинаризации и метода четырёхсвязности

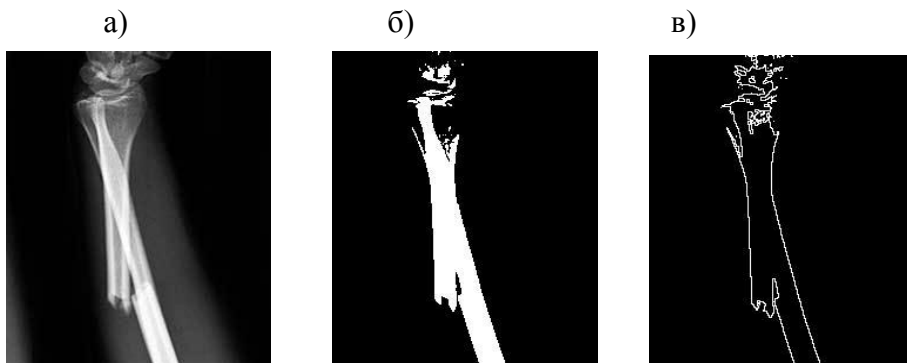


Рисунок -4 Результаты получения контура объекта на рентгенографическом изображении
а) исходное изображение,
б) бинаризованное изображение,
в) оконтуренное изображение.

Для исследования метода Канни и метода Собеля был использован программный пакет MATLAB. Результаты оконтуривания этих методов показаны на рисунке 5.

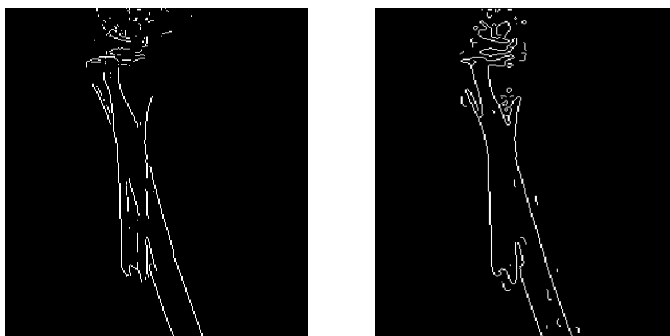


Рисунок -5 Оконтуренные изображения методом Канни и методом Собеля

Проведен анализ методов контурной сегментации для изображений рентгенограмм. На основе полученных результатов оконтуривания изображения можно сделать вывод, что среди исследуемых методов наиболее лучший – бинаризация по порогу

яркости с использованием четырёхсвязности, потому что этот метод, по сравнению с остальными, наиболее четко определяет контуры объекта.

Выводы

В статье исследованы методы обработки рентгеновских изображений. Были исследованы такие основные методы оконтуривания объекта как: метод Канни, метод Собеля, бинаризация по порогу яркости с использованием четырёхсвязности. Из выше перечисленных методов наиболее хорошо себя показал метод с использованием четырёхсвязности.

Литература

1. Функциональные элементы и устройства оптоэлектроники / Кожемяко В. П., Тимченко Л. И., Лысенко Г. Л., Кутаев Ю. Ф. – К.: УМК ВО, 1990. – 251 с.

2. Кормановський С. І. Підхід до визначення центру зв'язності зображення / Кормановський С. І., Швейкі Нафез, Тимченко Л. І. – Вісник ВПШ. – 2001. – № 4. – С. 71 – 73.

3. Canny J.E. A computational approach to edge detection //IEEE Trans Pattern Anal-ysis and Machine Intelligence. — 1986. — № 8. — P. 679—698.

4. Крылов В. Н., Полякова М. Морфологический метод контурной сегментации

изображений на основе репагулярного вейвлет-преобразования: Труды Одесского политехнического университета, г. Одесса, 2006. — С. 98—103.

5. Sobel I., Feldman G. «A 3x3 Isotropic Gradient Operator for Image Processing», 1968 (неопубл.).

6. Duda R., Hart P. Pattern Classification and Scene Analysis. — John Wiley and Sons, 1973. — P. 271—272.

УДК 001.891.32

А.И. Поляков, Д.Д. Новиков, А.А. Щербаков
Донецкий национальный технический университет
кафедра автоматизированных систем управления
Email: a_pai@ukr.net novikov.d92@mail.ru

АНАЛИЗ ВИДОВ ОПРЕДЕЛЕНИЯ ЧИСЛЕННОСТИ ГРУПП НАБЛЮДЕНИЯ ДЛЯ СТАТИСТИЧЕСКИХ РАСЧЕТОВ ПОКАЗАТЕЛЕЙ ОЦЕНКИ ЭФФЕКТИВНОСТИ ЛЕЧЕНИЯ БОЛЬНЫХ.

Аннотация.

Поляков А.И., Новиков Д.Д., Щербаков А.А. Анализ видов определения численности групп наблюдения для статистических расчетов показателей оценки эффективности лечения больных. Главной идеей статьи является поиск эффективного соотношения численности групп наблюдения за больными, который бы обеспечил минимальную общую стоимость исследования, без снижения его надежности.

Существует ряд причин научного, практического и этического характера, что заставляет исследователя заранее, еще в период организации и планирования контролируемого клинического испытания, решать вопрос о численности групп наблюдения, необходимой для обеспечения достоверных результатов.

Сравнивая два метода лечения, А и В, исследователь должен прийти к одному из трех выводов: а) метод А лучше, чем метод В; б) метод В лучше, чем метод А; в) полученные данные не позволяют судить о преимуществах того или иного метода лечения. Заключение, сделанные при сравнении двух групп больных, получавших разные виды лечения, подвержены риску различных ошибок:

1. Ложное утверждение о различиях между эффективностью сравниваемых методов лечения, которые на самом деле не различаются – ошибка первого рода с

вероятностью (α).

2. Ложное утверждение об отсутствии значимого различия между эффективностью сравниваемых методов лечения (не дается заключение), когда на самом деле они различаются – ошибка второго рода с вероятностью (β).

3. Ложное утверждение о преимуществе одного из методов лечения, когда в действительности большей эффективностью обладает сравниваемый метод - ошибка третьего рода с вероятностью (γ).

Если исследователь ориентирован в направлении эффекта одного из методов, учитывается односторонняя вероятность ошибочного утверждения того, например, что метод лечения В лучше метода А, исключая возможность превосходства последнего (односторонний тест). В данном случае проверяется гипотеза $\mu_B = \mu_A$ или $P_B = P_A$ против альтернативы $\mu_B > \mu_A$ или $P_B > P_A$, где μ_B и μ_A , P_B и P_A - истинные значения статических показателей эффекта (средних - μ и частот p) соответственно методов А и В.

При этом в формулу для определения числа наблюдений вводят значение нормального отклонения

$$Z = Z_{\alpha} + Z_{\beta}.$$

Когда интересует, существует ли различие в эффективности методов вообще (в том или противоположном направлении), т. е. проверяется гипотеза $\mu_B = \mu_A$ или $P_B = P_A$ против альтернативы $\mu_B \neq \mu_A$ или $P_B \neq P_A$, уровень ошибки первого рода α заменяется на $\alpha/2$ и учитывается ошибка третьего рода γ :

$$Z = Z_{\alpha/2} + Z_{\beta+\gamma} \text{ (двусторонний тест).}$$

Таким образом, как в одностороннем, так и в двустороннем текстах учет уровней ошибок играет первостепенную роль для определения необходимого числа наблюдений. Табл. 1 иллюстрирует возможные ситуации допущения различных ошибок [Schwartz D. et al.1980].

Таблица 1. Возможные варианты принятых решений и связанные с ними типы ошибок (двусторонний тест)[2]

Имеет место в действительности	Принятое решение		
	$B-A < 0$	$B-A = 0$ (нет заключения)	$B-A > 0$
$B-A = -\Delta$		Ошибка II рода β	Ошибка III рода γ
$B-A = 0$	Ошибка I рода $\alpha/2$		Ошибка I рода $\alpha/2$
$B-A = +\Delta$	Ошибка III рода γ	Ошибка II рода β	

При расчете необходимого числа наблюдений уровни и род ошибок устанавливаются в зависимости от типов данных исследований.

В фундаментальном исследовании, когда нас интересует, имеются ли различия в эффекте двух видов лечения, необходимо установить небольшое допустимое значение для ошибки I рода α и желательно для ошибки II рода β что достигается (при фиксированном α) достаточно большим числом наблюдений. При малых значениях α и β ошибка III рода γ становится пренебрежительно малой (если, например $B > A$, то маловероятно: чтобы наблюдающаяся разность оказалась значимой в противоположном направлении).

При практическом исследовании, когда интересует лишь, какой вид лечения рекомендовать, уровень ошибки I рода не имеет значения (если на самом деле $A = B$. то ничего не случится, какой бы из этих методов лечения мы ни рекомендовали) просто устанавливают на наибольшее допустимое значение, т.е. $\alpha = 100\%$. В практику рекомендуется метод который дает лучший наблюдающийся результат, т.е. всегда делается определенное заключение, что эквивалентно $\beta = \alpha$. Наиболее важной при практических клинических последствиях исследованиях является ошибка III рода γ - заключение с «неверным знаком» (см.табл.1), т.е. рекомендация худшего вида лечения.

Таким образом, для фундаментальных исследований устанавливаются малые уровни α и β при $\gamma=0$, а для практических — малый уровень γ , при этом фактически $\alpha=100\%$ и $\beta=0$. Вот почему фундаментальная оценка эффективности нового метода лечения и практического выбора лучшего из двух методов лечения требует разного числа наблюдений.

Численность групп наблюдения, обеспечивающая достоверность результатов контролируемого исследования, зависит от предполагаемых различий в эффекте лечения между опытной и контрольной группой (Δ), от заданной доверительной вероятности этих различий (т. е. от величины нормального отклонения Z), от вида статистического показателя эффекта лечения (частоты, средние) и от соотношения между числом больных в опытной (n_0) и контрольной (n_k) группах.

При создании статистической базы расчетов показателей оценки эффективности лечения целью является формирование предпосылок для выработки рекомендаций по практическому применению вида лечения, дающего лучший результат. При этом особенность оценки сопоставляемых методов лечения заключается в том, что все критерии, как непосредственно связанные с течением болезни, так и имеющие практическое значение (осложнения лечения, сложности его проведения или высокая стоимость и др.) рассматриваются вместе. Для такого сравнения существуют две возможности :

а) используется обобщенный критерий

$$x = a_1x_1 + a_2x_2 + \dots + a_kx_k,$$

где a_i отражает относительную важность критерия x_i ,

б) применяется один абсолютно доминирующий критерий x , отражающий прямые интересы больного (например, выживаемость), остальные критерии принимаются равными для обоих методов лечения или несущественными.

Сравниваемые методы лечения считаются эквивалентными, если они приведут к одинаковым значениям x , т. е. $\mu_B - \mu_A = 0$,

$$P_B - P_A = 0.$$

Минимальное число наблюдений в каждой группе больных при втором подходе определяется по следующим формулам:

$$\text{Когда известно } P_K \text{ и задается } P_0: n_0 = n_K = \frac{2Z_\gamma^2}{(\varphi_0 - \varphi_K)^2}$$

Рекомендуемые условия применения данной формулы $n > 20$.

$$\text{Когда известно } m_K \text{ и задается } m_0: n_0 = n_K = Z_\gamma^2 \frac{\sigma_0^2 + \sigma_K^2}{(m_0 - m_K)^2}$$

Рекомендуемые условия применения данной формулы $n > 30$, где n_0 и n_K - соответственно численность опытной и контрольной групп; Z_γ - нормальное отклонение, соответствующее установленному уровню ошибки III рода γ ., $\varphi = 2 \arcsin \sqrt{P}$, P и m - показатели лечебного эффекта, выраженные соответственно в частотах и средних величинах.

Рассматривая неравные по численности группы следует отметить, что одна и та же общая численность наблюдений $n = n_0 + n_K$ обеспечивает наибольшую точность результатов исследования при равном числе больных в опытной (n_0) и контрольной (n_K) группах, т. е. минимальное общее число наблюдений при заданной точности потребуется при равных группах. Однако бытующее мнение об обязательном соблюдении этого условия при контролируемых испытаниях ошибочно. Необходимость формирования разных по численности опытной и контрольной групп возникает как по ряду объективных причин, связанных главным образом с принимаемой во внимание «ценой» испытания (финансовой, этической и др.), так и субъективных. Исследователь судит о результатах испытания нового метода лечения не только путем сопоставления с аналогичными данными в контрольной группе, но и учитывая свой личный опыт наблюдений. С этих позиций, например, если нет каких-либо данных, ограничивающих, численность опытной группы, размеры ее можно увеличить за счет снижения численности

контрольной группы. Рассмотрим несколько вариантов:

а) Одному из изучаемых методов лечения может подвергнуться лишь определенное ограниченное число больных. Расчет такой численности второй группы наблюдения, которая обеспечила бы заданную достоверность исследования, если одно из значений n , например n_k , известно, то n_0 можно определить по формулам, приведенным в табл.2.

Таблица 2. Формулы расчета необходимого числа наблюдений при фиксированном n_k .

Вид исследования	Вид показателя	Односторонний тест [1]	Двусторонний тест
Фундаментальные	Средняя частота	$\frac{(Z_{\alpha} + Z_{\beta})^2}{(\varphi_0 - \varphi_k)^2 - \frac{(Z_{\alpha} + Z_{\beta})^2}{n_k}}$ $\frac{(Z_{\alpha} + Z_{\beta})^2}{(m_0 - m_k)^2 - \frac{(Z_{\alpha} + Z_{\beta})^2 \sigma_k^2}{n_k}}$	$\frac{(Z_{\alpha/2} + Z_{\beta})^2}{(\varphi_0 - \varphi_k)^2 - \frac{(Z_{\alpha/2} + Z_{\beta})^2}{n_k}}$ $\frac{(Z_{\alpha/2} + Z_{\beta})^2}{(m_0 - m_k)^2 - \frac{(Z_{\alpha/2} + Z_{\beta})^2 \sigma_k^2}{n_k}}$
Практические	Средняя частота		$\frac{Z_{\gamma}^2}{(\varphi_0 - \varphi_k)^2 - \frac{Z_{\gamma}^2}{n_k}}$ $\frac{Z_{\gamma}^2 \sigma_0^2}{(m_0 - m_k)^2 - \frac{Z_{\gamma}^2 \sigma_k^2}{n_k}}$

В приведенных формулах: Z_{α} , Z_{β} , Z_{γ} - нормальное отклонение, соответствующее установленным уровням ошибок I (α), II (β) и III (γ) рода; $\varphi = 2 \arcsin \sqrt{P}$; p и m - показатели лечебного эффекта, выраженные соответственно в частотах и средних величинах.

б) Применение одного из сопоставляемых методов лечения связано с определенным риском или техническими трудностями. Задача заключается в том, чтобы свести к минимуму эту группу больных, увеличив для сохранения заданной достоверности

численность второй группы в l раз. Например, тогда, если в формулах, приведенных в табл.2, заменить n_k на ln_0 то численность наблюдений в опытной группе при заданных l можно будет определить по формулам, представленным в табл.3, где $n_{l=1}$ необходимая численность при равномерном распределении больных по группам.

D.B. Owen показал, что если бы удалось увеличить одну из групп даже до бесконечности, то необходимое число наблюдений второй не уменьшилось бы более чем на $0.5n$, вычисленного при условии равенства групп ($l = 1$). Определить, на какую долю этого предела уменьшается число больных в одной из групп, можно по формуле: $100(1-1/l)$

При $l=2$ размер группы уменьшится на $100(1 - 1/2) = 50\%$ ее численности, установленной для равных групп; при $l=4$ на 75% и т. д.

Таблица 3. Формулы для расчета необходимого числа наблюдений n_0 при заданном соотношении между численностью

групп $\frac{n_k}{n_0} = l$.

Вид исследования	Вид показателя	Односторонний тест	Двусторонний тест
Фундаментальные	Средняя частота	$\frac{(Z_\alpha + Z_\beta)^2(l+1)}{l(\varphi_0 - \varphi_k)^2} =$ $= \frac{(l+1)n_{l=1}}{2l}$ $\frac{(Z_\alpha + Z_\beta)^2(\sigma_0^2 l + \sigma_k^2)}{l(m_0 - m_k)^2}$	$\frac{(Z_{\alpha/2} + Z_\beta)^2(l+1)}{l(\varphi_0 - \varphi_k)^2} =$ $= \frac{(l+1)n_{l=1}}{2l}$ $\frac{(Z_{\alpha/2} + Z_\beta)^2(\sigma_0^2 l + \sigma_k^2)}{l(m_0 - m_k)^2}$
Практические	Средняя частота		$\frac{Z_\gamma^2(l+1)}{l(\varphi_0 - \varphi_k)^2} =$ $= \frac{(l+1)n_{l=1}}{2l}$ $\frac{Z_\gamma^2(\sigma_0^2 l + \sigma_k^2)}{l(m_0 - m_k)^2}$

При планировании исследования нецелесообразно брать l более 4, так как при больших значениях l темп уменьшения лимитированной по численности группы резко снижается.

в) Стоимость одного из методов лечения очень высокая. Задача заключается в том, чтобы найти соотношение численности групп наблюдения (например, $n_{kl} = ln_0$), обеспечивающее минимальную общую стоимость исследования без снижения его надежности.

Для этой цели может быть использовано «правило квадратного корня», теоретически обоснованное W. G: Cochran (1963), O. Miettien (1969), J. Nam (1973).

В несколько преобразованном виде оно может быть сформулировано следующим образом: если стоимость курса лечения в одной из групп в r раз выше, чем в другой, а отношение

$$\frac{\sigma_2^2}{\sigma_1^2} = \frac{p_2q_2}{p_1q_1} = d,$$

где $q = 100 - P$, то чтобы свести к минимуму общую стоимость испытания, сохранив достоверность различий в эффекте на том же уровне, необходимо в каждой из групп взять такое число наблюдений, чтобы на одного больного группы n_1 приходилось \sqrt{rd} больных в группе n_2 , т. е. $l = \sqrt{rd}$.

Подставляя эти значения l и d в формулы вычисления численности одной группы по заданной численности другой, получим следующие формулы (табл.4).

г) один из сопоставляемых лечения причиняет определенные неудобства больным, которые можно выразить количественно. При этом подходе задача заключается в том, чтобы подвергнуть такому методу лечения меньше больных, сохранив при этом заданную достоверность исследования.

Таблица 4. Формулы расчета необходимого числа наблюдений n_0 ($l = \sqrt{rd}$)

Вид исследования	Вид показателя	Односторонний тест	Двусторонний тест
Фундаментальные	Средняя частота	$\frac{(Z_{\alpha} + Z_{\beta})^2 (\sqrt{rd} + 1)}{\sqrt{rd} (\varphi_0 - \varphi_k)^2} =$ $= \frac{(\sqrt{rd} + 1)n_{l=1}}{2\sqrt{rd}}$ $\frac{(Z_{\alpha} + Z_{\beta})^2 (\sqrt{rd} + d)\sigma_0^2}{\sqrt{rd} (m_0 - m_k)^2}$	$\frac{(Z_{\alpha/2} + Z_{\beta})^2 (\sqrt{rd} + 1)}{\sqrt{rd} (\varphi_0 - \varphi_k)^2} =$ $= \frac{(\sqrt{rd} + 1)n_{l=1}}{2\sqrt{rd}}$ $\frac{(Z_{\alpha/2} + Z_{\beta})^2 (\sqrt{rd} + d)\sigma_0^2}{\sqrt{rd} (m_0 - m_k)^2}$
Практические	Средняя частота		$\frac{Z_{\gamma}^2 (\sqrt{rd} + 1)}{\sqrt{rd} (\varphi_0 - \varphi_k)^2} =$ $= \frac{(\sqrt{rd} + 1)n_{l=1}}{2\sqrt{rd}}$ $\frac{Z_{\gamma}^2 (\sqrt{rd} + d)\sigma_0^2}{\sqrt{rd} (m_0 - m_k)^2}$

Если эти «неудобства» условно принять за «стоимость», то можно воспользоваться «правилом квадратного корня».

Рассмотренные методы определения минимального числа наблюдений обеспечивают достоверность результатов при заданных условиях, т.е. при уровнях вероятностей ошибок (α , β , γ) и предлагаемых различиях в эффекте по определенному критерию (Δ).

Список литературы

1. Сергиенко В.И., Бондарева И.Б. Математическая статистика в клинических исследованиях. – М.: ГЭОТАР-МЕД, 2001. – 256 с.
2. Anthony J Culyer, B. Horisberger Economic and Medical Evaluation of Health Care Technologies, 1983г 395с.

УДК 004.045:004.942

ИСПОЛЬЗОВАНИЕ МЕТОДОВ ПОИСКА ОПТИМАЛЬНОГО ПУТИ ПРИ ПОСТРОЕНИИ МАРШРУТА ДЛЯ ПРОВЕДЕНИЯ МОНИТОРИНГА ВОДНОЙ СРЕДЫ

Сафонов М.Д., Светличная В.А.

Донецкий национальный технический университет г.Донецк
кафедра автоматизированных систем управления
e-mail: msafonovmail@gmail.com

Аннотация

Сафонов М.Д., Светличная В.А. Использование методов поиска оптимального пути при построении маршрута для проведения мониторинга водной среды. В данной статье описывается реализация методов для определения наилучшего пути исследовательских судов, проводящих измерения параметров водной среды, с помощью генетического алгоритма, метода ветвей и границ и муравьиного алгоритма

Постановка задачи.

Основной задачей исследований водной среды является изучение пространственно-временной изменчивости параметров океана. Изменчивость характеристик водной среды в пространстве изучается с помощью различной зондирующей и буксируемой аппаратуры, устанавливаемой на надводных и подводных кораблях, подводных аппаратах. [1].

Средства и методы исследования океана являются одной из быстро развивающихся отраслей современной океанологии. С одной стороны, это вызвано потребностями самой океанологической науки. С другой стороны, ряд важных и проблемных прикладных народно-хозяйственных задач стимулирует разработку различной аппаратуры.

Для дальнейшего изучения и освоения океана, необходимо регулярно получать данные о его структуре и состоянии. Вместе с тем измерение параметров характеризующих состояние водной среды довольно трудная и дорогая операция. Поэтому одной из основных задач является построение наиболее эффективных алгоритмов и методов для измерения параметров водной среды.

Постановка задач исследования.

Экспедиционные измерения гидрофизических параметров - дорогостоящая операция. Одним из слабых мест систем, производящих измерения, является не оптимальный маршрут при экспедиционных исследованиях. Использование при построении маршрута корабля опорных точек, позволит уменьшить затраты на измерения.

Поэтому необходимо проводить измерения по возможности оптимальным путем. Исходя из этого, одной из основных целей информационных измерительных систем является - оптимизация маршрута экспедиционного корабля. На данный момент для экспедиционных исследований задана акватория разбивается на квадраты. Корабль проходит поочередно все квадраты акватории. Примерная схема маршрута изображена на рисунке 1.

Таким образом, экспедиционное судно получает наиболее полные данные о заданной области. Однако они обычно количество данных является избыточным.

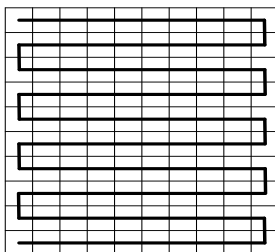


Рисунок 1. - Маршрут экспедиционного корабля

Функция изменения измеряемого параметра обладает экстремумами в заданной области. То есть для данного параметра можно найти несколько максимальных и минимальных значений.[2,3]

Набор значений измеренных между этими точками, дает представление о градиенте изменения исследуемого параметра в данной области. Из этого следует, что выполнив измерение значения параметров в точках, соответствующих экстремумам, а также между ними, возможно, сформировать полную картину о распределении заданного параметра в акватории. Исходя из этого, возможно определенным образом оптимизировать маршрут экспедиционного судна, то есть производить измерения не в каждом квадрате, а в квадратах, соответствующих расположению опорных точек.. Вопрос нахождения опорных точек, который рассматривался в [], является первой частью общей задачи. Определение наиболее рационального маршрута для обхода опорных точек представляет ее вторую часть.

Решение задачи и результаты исследований.

Задачу поиска оптимального пути по опорным точкам можно формализовать как задачу коммивояжера, которая является одной из самых известных задач комбинаторной оптимизации. Задача заключается в отыскании самого выгодного маршрута, проходящего через указанные города хотя бы по одному разу с последующим возвратом в исходный город. В условиях задачи указываются критерий выгодности маршрута (кратчайший, самый дешевый, совокупный критерий и т.п.) и соответствующие матрицы расстояний, стоимости и т.п. Как правило, указывается, что маршрут должен проходить через каждый город только один раз - в таком случае выбор осуществляется среди гамильтоновых циклов. Для рассматриваемой задачи построение наиболее оптимального пути корабля с измерительной системой выполняется по стоимости.

Существует несколько частных случаев общей постановки задачи, в частности геометрическая задача коммивояжера (также называемая планарной или евклидовой, когда матрица расстояний

отражает расстояния между точками на плоскости), треугольная задача коммивояжера (когда на матрице стоимостей выполняется неравенство треугольника), симметричная и асимметричная задачи коммивояжера. Также существует обобщение задачи, так называемая обобщенная задача коммивояжера [4].

Общая постановка задачи, как и большинство ее частных случаев, относится к классу NP-Сложных задач. Все эффективные (сокращающие полный перебор) методы решения задачи коммивояжера - методы эвристические.

В большинстве эвристических методов находится не самый эффективный маршрут, а приближенное решение. Чаще востребованы так называемые *any-time* алгоритмы, то есть постепенно улучшающие некоторое текущее приближенное решение.

Существует также постановки, в которых задача коммивояжера становится NP-Полной, Обычно такие постановки выглядят следующим образом: существует на заданном графе G такой обход, что его стоимость не превышает x . Часто на ней проводят обкатку новых подходов к эвристическому сокращению полного перебора. На практике используются различные модификации наиболее эффективных методов: метод ветвей и границ, метод генетических алгоритмов, а также метод муравьиных колоний.

Для поиска оптимального маршрута было рассмотрены три метода: ветвей и границ, генетический алгоритм и алгоритм муравьиной колонии. Для определения наиболее подходящего для данной задачи метода были реализованы все. Представим описание объектных моделей данных методов.

Метод ветвей и границ

При реализации метода ветвей и границ были использованы следующие классы:

1) `RouteAlgorithm` - интерфейс определяющий функциональность класса реализующего построение оптимального маршрута.

2) **BranchBordersAlgorithm** – основной класс модели реализующий интерфейс **RouteAlgorithm**. В данном классе заполняется матрица стоимостей переходов между узлами и производится поиск наиболее оптимального маршрута.

3) **Subset** – класс определяющий подмножество. В данном классе реализована редукция матрицы стоимостей, поиск текущей наиболее выгодной дуги, а также создание новых матриц. Каждое подмножество ссылается на своего родителя, для возможности возврата вверх по графу и проверки на наличие более оптимального решения.

4) **Route** - класс определяющий маршрут. Хранит в себе последовательность подмножеств образованных в ходе выполнения алгоритма. Выполняет преобразование набора подмножеств в маршрут, а также проверку полученного маршрута на незамкнутость.

5)

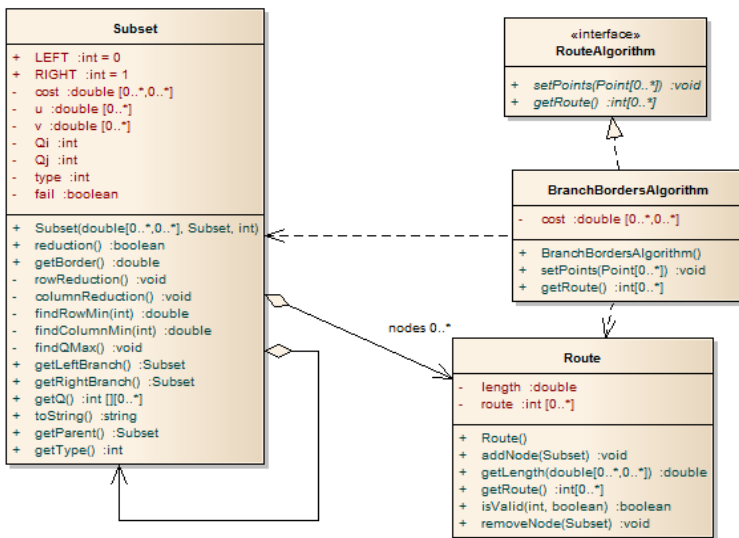


Рисунок 2 – Диаграмма классов метода ветвей и границ

Генетический алгоритм.

Для определения наиболее подходящего метода было реализовано три представления хромосомы в генетическом алгоритме: путевое, соседство и порядковое. Для каждого представления были реализованы свои варианты кроссинговера. Рассмотрим классы созданные при реализации генетического алгоритма:

1) `RouteAlgorithm` - интерфейс определяющий функциональность класса реализующего построение оптимального маршрута.

2) `GenAlgorithm` – класс реализующий интерфейс `RouteAlgorithm`. Производит инициализацию системы, а также пошаговое выполнение эволюционного алгоритма до выполнения условия останова. Останов алгоритма происходит, когда разница между средним значением длины маршрута и минимальным значением не превышает заданный порог.

3) `Population` – класс реализующий популяцию. Хранит в себе набор хромосом заданного типа. Реализует операции репродукции, скрещивания, мутации и отсева хромосом.

4) `Chromosome` - интерфейс определяющий функциональность хромосомы при решении задачи коммивояжера.

5) `NeighbourViewChromosome` – абстрактный класс реализующий интерфейс `Chromosome`. Реализует представление соседство. Генерирует хромосому в таком представлении, проверяет на корректность, а также преобразовывает хромосому в список узлов. В зависимости от операции кроссинговера расширяется двумя классами – `HeuristicChromosome`, `AlternatingEdgesChromosome`.

6) `SerialViewChromosome` – класс реализующий интерфейс `Chromosome`. Реализует порядковое представление. Генерирует хромосому в таком представлении, проверяет на корректность, а также преобразовывает хромосому в списке узлов.

7) `RouteViewChromosome` – абстрактный класс реализующий интерфейс `Chromosome`. Реализует путевое

представление . Генерирует хромосому в таком представлении, проверяет на корректность. В зависимости от операции кроссинговера расширяется тремя классами – PartiallyMappedChromosome, CircleChromosome, OrderChromosome.

8) ChromosomeFactory – класс-фабрика. Создает объект хромосома заданного типа.

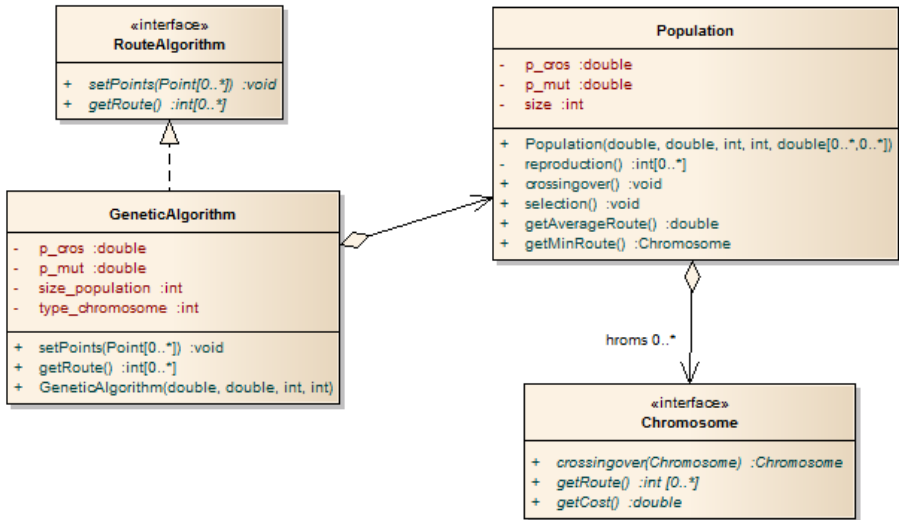


Рисунок 3 – Диаграмма классов генетического алгоритма

Муравьиный алгоритм

В ходе реализации муравьиного алгоритма для решения задачи построения оптимального маршрута были разработаны следующие классы:

1) RouteAlgorithm – интерфейс определяющий функциональность класса реализующего построение оптимального маршрута.

2) AntAlgorithm – класс реализующий интерфейс RouteAlgorithm. Класс инициализирует систему построения оптимального маршрута. Также класс пошагово выполняет

прохождение муравьями выбранных маршрутов и обновление количества феромона.

3) **Ant** – класс реализующий функции муравья. Хранит список пройденных узлов, а также определяет вероятностным способом следующую дугу по которой муравей продолжит маршрут. По окончании пути муравей оставляет след феромона на всех пройденных дугах пропорциональный силе феромона и обратно пропорциональный длине пройденного маршрута .

4) **Arc** - представляет одну из дуг соединяющих два узла. Дуга хранит свою длину, а также количество феромона. На каждом шаге количество феромона уменьшается в соответствии с указанным коэффициентом.

5) **Node** - класс реализующий узел маршрута. Содержит список дуг по которым можно выйти из города, а также список муравьев которые находятся на данном шаге в том узле.

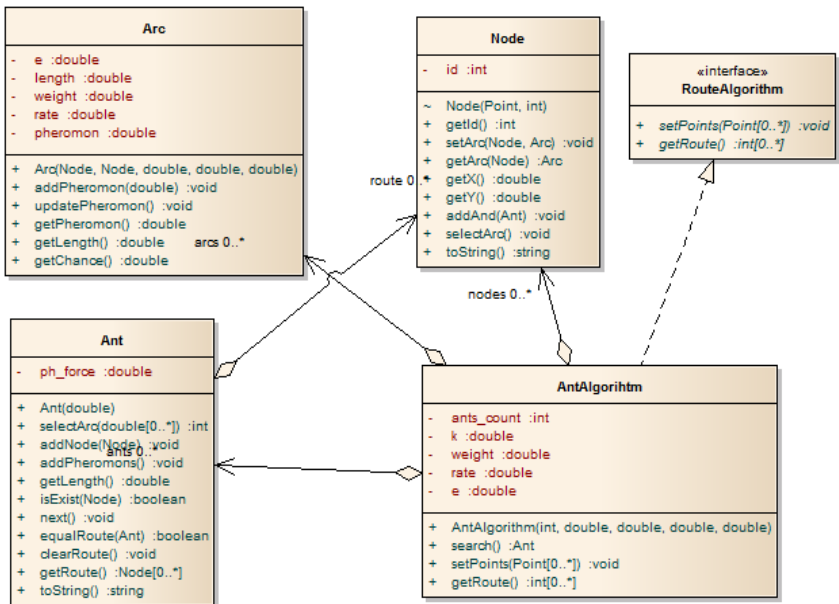


Рисунок 4 – Диаграмма классов муравьиного алгоритма

Выводы.

В результате были рассмотрены три алгоритма поиска оптимального маршрута - генетический алгоритм, метод ветвей и границ и муравьиный алгоритм. При проверке алгоритмов на разных вариантах расположения опорных точек представляется получать разные предполагаемые оптимальные маршруты.

Список литературы

1. Зори А.А., Коренев В.Д., Хламов М.Г. Методы, средства, системы измерения и контроля параметров водных сред. - Донецк: РИА ДонГТУ, 2000. – 388с.
2. Чесноков В. І., Світлична В. А., Савкова О. Й. Визначення опорних точок у процесі планування моніторингу водяного середовища. - Міжвідомчий збірник наукових праць «Відбір і обробка інформації» Випуск 33(109), 2011 – с. 61-67
3. Савкова Е.О., Светличная В.А. Объектно-эволюционная модель системы мониторинга гидрофизических полей водной среды. - Наукові праці Донецького національного університету Серія: Обчислювальна техніка та автоматизація» Випуск 171(19) – Донецьк:ДонНТУ, 2010 – с.224.
4. Новиков Ф.А Дискретная математика для программистов –С.Петербург изд. Дом. «Питер».2002 – 304с.

ИССЛЕДОВАНИЕ МЕТОДОВ ОПТИМИЗАЦИИ СТРУКТУРЫ И КОНТЕНТА САЙТА НА ОСНОВАНИИ ДАННЫХ АНАЛИТИКИ

Чепикова Е.Д., Савкова Е.О.

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления

Аннотация

Чепикова Е.Д., Савкова Е.О. Исследование методов оптимизации структуры и контента сайта на основании данных аналитики. Выполнен анализ методов оптимизации контента. Выбрана последовательность методик для генерации динамического контента. Разработана модель программного решения создания контента на основе данных веб-аналитики.

Ключевые слова: методы оптимизации контента сайта, веб-аналитика, динамическая генерация контента.

Постановка проблемы.

В данный момент, поисковые системы продолжают развиваться и, в скором времени, смогут ранжировать сайты, принимая во внимание, популярность контента. То есть сайты, с «непопулярным» содержанием будут опускаться ниже в результатах поискового запроса. А значит, сайты должны максимально подходить под потребности пользователей веб-ресурса. Как следствие, возникает необходимость подбирать контент под предпочтения пользователей. Для реализации этого нужно выполнить следующие этапы:

- собирать данные о переходах пользователей, используя веб-аналитику;
- проводить анализ данных на посещённых страницах;
- формировать динамический контент с учётом предыдущих посещений.

Цель статьи. Для оптимизации сайта самое важное – это конвертировать полученные данные в практические действия по оптимизации контента. Готовить контент только с учетом «требований» поисковых систем будет не совсем правильно. Ориентироваться надо на подготовку актуального, интересного и полезного контента. [1].

В данной работе целью является проведение анализа методов оптимизации контента и отбор и их обобщение для реализации самой оптимизации.

Постановка задачи исследования. В данной работе необходимо определить используемый для создания веб-системы с медиа-контентом метод для оптимизации его структуры и содержимого. А также построить свою модель системы оптимизации экспертной системы.

Решение задач и результаты исследований.

На веб-сайте основным "коммуникатором" и "путеводителем" для посетителя является информационный контент. Если контент не захватывает внимание посетителя, не мотивирует его на те или иные действия - то вне зависимости от уровня посещаемости веб-сайта, он не будет эффективно выполнять свои функции [1].

Для определения лучшего метода оптимизации, была исследована поисковая оптимизация.

Поисковая оптимизация сайта предполагает большое количество текстов на сайте, содержащих ключевые запросы. Вхождение ключа – употребление ключевого слова в сегменте текста. Обычно ключи располагают подальше друг от друга, чтобы их плотность в тексте была оптимальной. Также необходим текст на главную страницу сайта – 2-3 тысячи знаков, содержащий ключевые запросы по максимуму, т.е. в полном объеме.

Задача поисковой оптимизации заключается в формировании такого ядра запросов (ключей), которые были бы одинаково удобны для данной поисковой системы (обработка запросов

поисковыми машинами) и пользователей интернет, которые эти запросы вводят.

Любой сайт содержит ключи первого уровня и ключи второго уровня (сопутствующие запросы) – задача поисковой оптимизации заключается ещё и в том, чтобы вывести сайт в лидеры в строках запросов, как для первых, так и для вторых. [2].

Однако её сейчас оказывается недостаточно, о чём говорится в книге Дмитрия Мелихова. Т

ак как увеличение посещаемости сайта – это еще не значит повышение его конверсии [3]. Предлагается использование систем веб-аналитики. Они позволяют определить основные тенденции взаимодействия посетителя с веб-сайтом, выявить недостатки в деятельности сайта или отдельных его элементов [3].

Собственно говоря, веб-аналитика – это инструмент анализа и выявления проблем веб-ресурса, это не простая оптимизация сайта, а оптимизация под конкретные бизнес-процесс и задачи, это не перечень данных, а перечень мер по улучшению показателей эффективности ресурса.

Веб-аналитику используют решения следующих проблем:

- развитие функциональности и функционала сайта по результатам мониторинга поведения посетителей;
- организация эффективной рекламной кампании в интернет;
- выявление недостатков и путей их исправления на сайте (структура, навигация, дизайн, контент, оптимизация) [4].

Что касается анализа текстового контента веб-сайта, то его можно проводить практически с помощью:

- видеозаписи движения мыши посетителей на веб-страницах;
- тепловых карт взаимодействия посетителей с контентом;
- карт скроллинга веб-страницы с текстовым контентом;
- функции «локально сохраненные страницы» [3].

Ещё одним методом оценки качества контента является привлечение профессионального копирайтера, который с позиции своего опыта сможет сказать, насколько качественно

написан контент и нуждается ли он в доработке. Однако каждый копирайтер имеет свою, субъективную точку зрения, и нет таких специалистов, которые бы одинаково написали контент на одну и ту же тему [1].

Группа веб-аналитиков Яндекса пришла к выводу, что наиболее объективным будет использование веб-аналитики, чем привлечение копирайтера.

В связи с этим можно заявить, что для оптимизации сайта следует выбрать веб-аналитику.

Разрабатываемая система также будет использовать данную методику, однако она будет учитывать и запоминать переходы для каждого пользователя в отдельности.

Для привлечения пользователей на сайт необходимо предлагать пользователям фильмы, согласуясь с тем контентом, что их уже заинтересовал. А именно, отслеживать и сохранять адреса посещённых страниц пользователей. Затем проводить анализ фильмов, находящихся на этих страницах (на основе присущих им критериев), и формировать контент предлагаемых фильмов, а также, оптимизировать в соответствии с критериями поисковые запросы пользователей, выдвигая вперёд, те фильмы, которые были бы интересны именно ему, с наибольшим совпадением критериев.

Такой метод сможет привлечь большую аудиторию для сайта, так как обеспечивает удобство для посетителя и экономит его время.

Следовательно, для решения поставленной задачи предлагается исследовать применение следующей последовательности методов: аналитика по отслеживанию переходов на сайте; сбор статистики по посещённым страницам, анализ содержимого критериев фильмов, расположенных на данных страницах.

Укрупнённая модель алгоритма такой системы представлена на рис. 1.

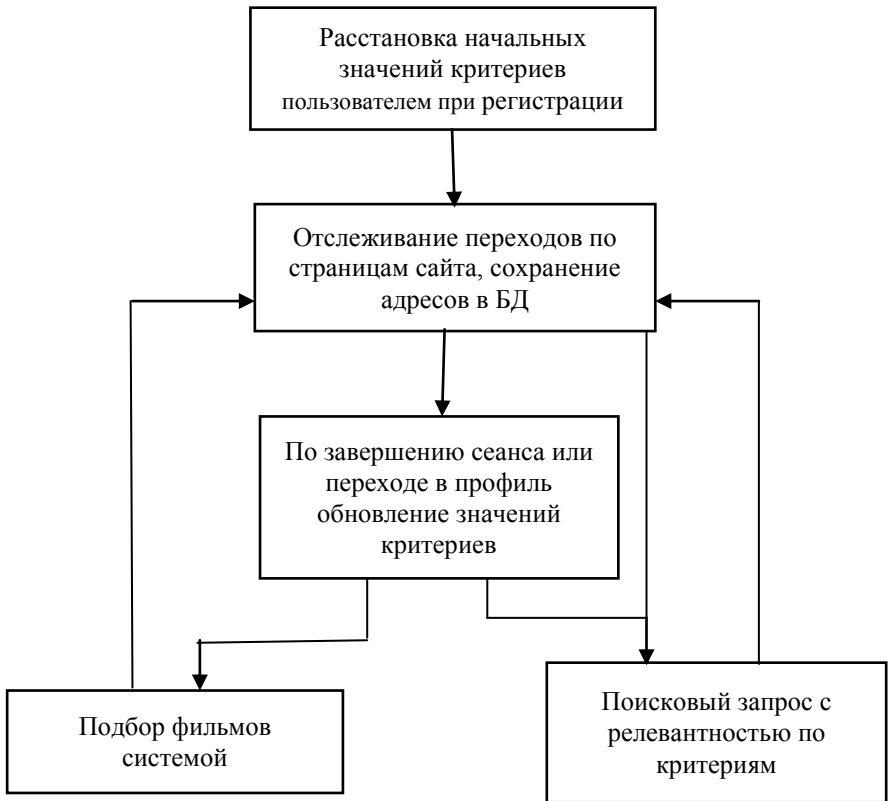


Рисунок 1 – обобщённая модель системы

Согласно разрабатываемой модели у каждого пользователя и фильма содержится список критериев. Для фильма он принимает значение true или false, присутствует он в фильме или нет. У пользователя критерий задан в виде рейтинга значимости, который увеличивается при посещении ссылок с фильмом, у которого этот критерий присутствует.

В профиле пользователя будет доступна возможность просмотра подобранных системой фильмов лично данному

пользователю. Подбор осуществляется таким образом. Для пользователя значимыми будут критерии с рейтингом выше 7 включительно (рейтинг от 0 до 10). Подсчитывается для фильма количество подходящих критериев (если значимы для пользователя критерий в фильме присутствует, то количество увеличивается на один). Затем список сортируется по убыванию количества критериев и ставится лимит на 100 записей.

При поисковом запросе, аналогичным образом подсчёта значимых критериев результаты располагаются по релевантности.

Выводы. Проведен анализ методов оптимизации контента. Для системы будет использоваться совмещение метода веб-аналитики и сбора статистики о предпочтениях пользователей. Описан алгоритм действия такой экспертной системы оптимизации структуры и контента, а также построена обобщённая модель системы. Следующими шагами работы должен стать выбор критериев для фильмов и разработка подробного алгоритма формирования динамического контента, с учётом критериев, которые нравятся пользователю.

Список литературы

1. Игорь Сарматов, Веб-аналитика: шаг к совершенству – К, 2010 -112 с.
2. Дмитрий Мелихов – Веб-аналитика: практика, теория, практика. - К, 2010 -26 с.
3. Поисковая оптимизация и продвижение сайта [Электронный ресурс] – Режим доступа: <http://mirsovetov.ru/a/hitech/network/seo-site-promotion.html> (дата обращения: 03.05.2014).
4. Веб-аналитика [Электронный ресурс] – Режим доступа: <http://allforjoomla.ru/info/304-veb-analitika> (дата обращения: 07.05.2014)
5. Методы поисковой оптимизации сайта [Электронный ресурс] – Режим доступа: <http://www.antula.ru/optimization-methods.htm> (дата обращения: 03.05.2014).



**КОМПЬЮТЕРНЫЙ
МОНИТОРИНГ И МЕТОДЫ
ИНТЕЛЕКТУАЛЬНОГО
АНАЛИЗА ДАННЫХ**

СЕКЦИЯ 2

УДК 004.67

А.В. Григорьев, Е.А. Заплетин

Донецкий Национальный Технический Университет
кафедра прикладной математики и информатики
E-mail: zapletin.yevhenii@gmail.com

ОСНОВНЫЕ ПОДХОДЫ ПРИ СОЗДАНИИ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ

Аннотация

Григорьев А.В., Заплетин Е.А. Основные подходы при создании рекомендательных систем. Рассмотрена классификация подходов для создания рекомендательных систем. Определено назначение и преимущества каждого из подходов. Определены основные проблемы при реализации данных подходов.

Общая постановка проблемы

Основная задача рекомендательных алгоритмов: анализировать запросы пользователя и на основе этих данных предугадывать его действия.

Наиболее простой способ рассмотреть основную проблему, которую решают алгоритмы рекомендаций можно на примере. В сервисах которые предлагают большой выбор товаров или контента существует проблема длинного хвоста [1]. Данная проблема подходит под категорию распределение Парето (отношение 20% к 80%). Так например 20% видов товаров в интернет-магазинах делают 80% выручки. Пользователи покупают в основном популярные товары, а про существование оставшихся 80% товаров пользователь может даже не знать. Таким образом появляется проблема, когда пользователь может не знать о существовании товара только потому, что он не популярен среди большинства пользователей. Для решения данной проблемы созданы алгоритмы рекомендаций которые персонализируют предложения исходя из данных о пользователях.

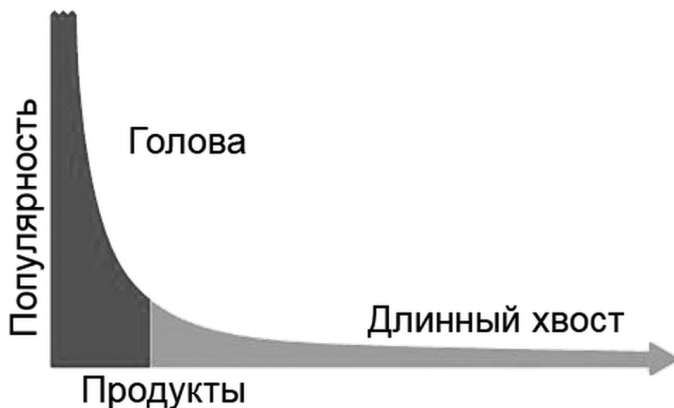


Рис. 1. Эффект длинного хвоста.

Исследования

Существует несколько типов рекомендационных систем[2]:

- Системы основанные на данных о действиях пользователей (collaborative systems);
- Системы основанные на данных о свойствах объектов рекомендаций и свойствах пользователей (content-based systems);
- Системы основанные на знаниях связанных с объектом рекомендаций (knowledge-based systems);
- Гибридные системы (hybrid systems).

1. Системы, основанные на данных о действиях пользователей

Системы используют только информацию о действиях пользователя. Основная идея следующая - если пользователи U1 и U2 купили книгу B1, а пользователь U2 купил еще и книгу B2, то логично предположить что книга B2 будет также интересна пользователю B1.

Стоит заметить, что данные системы не учитывают знания и данные про свойства пользователей и предметов рекомендаций. То есть неважно какого именно жанра будут купленные книги и также не имеет значения что пользователь указал в своем профиле что ему интересны книги определенного автора.

Плюсы:

- Нет необходимости выявлять свойства объектов;
- Нет необходимости обратной связи от пользователя;
- Обучения на множествах данных без участия человека;
- Нет разницы в предсказании любых типов объектов;
- Учитывается автоматическая обратная связь от пользователей(просмотры, установки и т.д.).

Минусы:

- “Холодный старт” для новых пользователей. Алгоритм будет работать неверно для новых пользователей, информации о которых очень мало;
- “Холодный старт” для новых предметов. Алгоритм будет работать неверно для новых предметов, информации о которых очень мало;
- Необходимость обрабатывать наибольшие массивы данных по сравнению с другими системами;
- Интересы пользователей усредняются;
- Рекомендации не основываются на интересах конкретного пользователя;
- Рекомендует в основном популярные объекты;
- Сложно указать причину рекомендации.

2. Системы, основанные на данных о свойствах объектов рекомендаций и свойствах пользователей

Системы данного типа не используют данные о действиях пользователя и только анализируют свойства рекомендуемых предметов и свойства текущего пользователя.

Данные свойства могут быть введены вручную(например пользователь указывает какой жанр книг ему интересен или администратор сайта вводит информацию про книгу: жанр, автор,

год издания, и т.д.), автоматически извлекаться из рекомендуемых предметов(например анализ длины текстов в книгах и разделения их на группы: короткие, средние, длинные) и нахождения новых признаков при помощи алгоритмов кластеризации анализируя исходные данные.

Плюсы:

- Нет “холодного старта” для новых предметов. Новые предметы сразу участвуют в процессе рекомендаций.
- Рекомендует непопулярные объекты;
- Рекомендации основываются на интересах конкретного пользователя;
- Легко указать причину рекомендаций;
- Можно использовать алгоритмы кластеризации для поиска новых свойств объектов.

Минусы:

- “Холодный старт” для новых пользователей. Алгоритм будет работать неверно для новых пользователей, информации о которых очень мало;
- Необходимо участие человека для выявления свойств объектов;
- Интересы пользователей усредняются;
- Желательно явное указание пользователем своих интересов;
- Не используется обратная связь от пользователей(оценки, нравится/не нравится, установки, просмотры и т.д.).

3. Системы, основанные на знаниях связанных с объектом рекомендаций

Системы из этой группы используют знания про сферу предполагаемых продуктов. Данные системы очень похожи на системы основанные на свойствах пользователя и предметов рекомендаций, и в некоторых источниках данный тип систем является подтипом систем основанных на свойствах.

Основной принцип работы данных систем основан на активном взаимодействии с пользователем(получения обратной

связи) и знаниях о сферах рекомендации. Данный тип систем используют в том случае, когда тип 1 и 2 использовать невозможно из-за небольшого количества данных. Примером может служить система рекомендации фотоаппаратов. Как правило, люди покупают новый фотоаппарат только раз в несколько лет, поэтому у каждого конкретного магазина будет недостаточно данных для рекомендации на основе действий пользователя. Также фотоаппараты имеют большое количество характеристик, которые имеют различную важность для пользователей. Для фотографов которые снимают в студии наиболее важно качество съемки, в тоже время для фотографов которые снимают на улице и ведут активный образ жизни также важен вес фотоаппарата, удобство переноски, размер памяти и время автономной работы.

Система рекомендаций данного типа активно взаимодействует с пользователем, но не просит ввести все параметры пользователя вручную. В примере с фотоаппаратами система будет спрашивать пользователя где будет происходит съемка, портретную или пейзажную съемку предпочитает пользователь. Анализируя ответы пользователя система определить наиболее важные для данного пользователя свойства и их оптимальные значения. Знания про сферу рекомендаций изначальной устанавливает администратор системы.

Плюсы:

- Нет “холодного старта” для новых предметов. Новые предметы сразу участвуют в процессе рекомендаций.
- Нет “холодного старта” для новых пользователей. Новые пользователи сразу получают правильные рекомендации.
- Рекомендует непопулярные объекты;
- Рекомендации основываются на интересах конкретного пользователя;
- Легко указать причину рекомендаций;
- Рекомендации учитывают интересы конкретного пользователя;
- Интересы пользователей не усредняются.

Минусы:

- Очень сложно и трудозатратно выделить знания о предметах;
- Добавления новых объектов требует участия человека;
- Практически нет автоматизации в обучении системы;
- Необходимо много обратной связи от пользователя;
- Желательно явное указание пользователем своих интересов;

4. Гибридные системы

Приведенные выше системы используют различные подходы к анализу данных и предсказанию результата, тем самым решая задачу различными способами. На практике, не один из способов в одиночку не может рекомендовать товары или контент с большой эффективностью. Поэтому реальные системы рекомендаций всегда используют несколько подходов к анализу и к рекомендациям.

Гибридные системы объединяют все описанные подходы в единый алгоритм и имеют наибольшую эффективность. Также данные системы наиболее сложны в реализации и проектировании. Сложность заключается в адаптации алгоритмов к конкретной сфере применения рекомендационной системы. Так например в сферах где каждый пользователь сам оценивает предлагаемый контент (каталог фильмов с оценками) наиболее важной частью являются оценки пользователей и анализ их действий. А при использовании рекомендательной системы на сайте интернет магазинов наиболее важными задачами для администратора системы является увеличение прибыли, а значит при создании рекомендации необходимо учитывать фактор цены товаров и необходимости их продать.

Плюсы:

- Нет “холодного старта” для новых предметов. Новые предметы сразу участвуют в процессе рекомендаций.
- Нет “холодного старта” для новых пользователей. Новые пользователи сразу получают правильные рекомендации.
- Рекомендует непопулярные объекты;

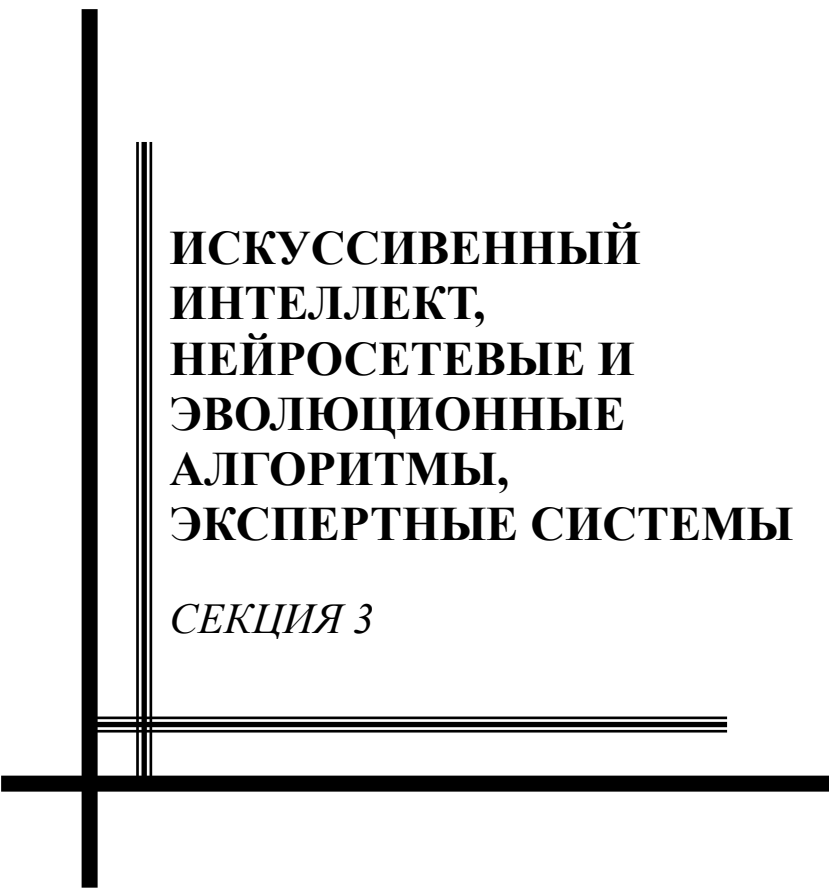
- Рекомендации основываются на интересах конкретного пользователя;
 - Рекомендации учитывают интересы конкретного пользователя;
 - Интересы пользователей не усредняются;
 - Наиболее эффективны.
- Минусы:
- Большая сложность в реализации;
 - Необходимо объединять воедино несколько алгоритмов в один;
 - Очень сложно и трудозатратно выделить знания о предметах;
 - Сложно указать причину рекомендаций;
 - Интересы пользователей усредняются.
 - Желательно обратная связь от пользователей;
 - Желательно участие человека при добавлении новых товаров.

Вывод

Существует несколько способов реализации алгоритма рекомендаций. Однако не один из способов в одиночку не может дать хороших результатов. Поэтому на практике всегда необходимо использовать гибридную систему, которая объединяет несколько способов.

Список литературы

1. Leskovec J. Mining of Massive Datasets [Текст] / J.Leskovec, A. Rajaraman, J. Ullman // Cambridge University Press. – Cambridge : 2011.
2. Francesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springe



**ИСКУССВИЕННЫЙ
ИНТЕЛЛЕКТ,
НЕЙРОСЕТЕВЫЕ И
ЭВОЛЮЦИОННЫЕ
АЛГОРИТМЫ,
ЭКСПЕРТНЫЕ СИСТЕМЫ**

СЕКЦИЯ 3

УДК 004.853

В.В. Елифёров, О.И. Федяев

Донецкий национальный технический университет

Кафедра прикладной математики и информатики

eliferov.vlad@mail.ru

ФОРМИРОВАНИЕ ОБУЧАЕЩЕГО МНОЖЕСТВА ЗАВИСИМОСТИ ЗНАНИЙ СТУДЕНТОВ ОТ ИХ МЕНТАЛИТЕТА ДЛЯ ДВУХСЛОЙНЫХ НЕЙРОННЫХ СЕТЕЙ

Елифёров В.В., Федяев О.И.. Формирование обучающего множества зависимости знаний студентов от их менталитета для двухслойных нейронных сетей. Данная нейросетевая модель имитирует процесс передачи профессиональных навыков и знаний в зависимости от личностных характеристик студентов, что позволит прогнозировать качество профессиональной подготовки по отдельной дисциплине уже на начальных этапах обучения. Программный комплекс позволит в целом анализировать и улучшать образовательный процесс.

Введение.

В качестве объекта исследования рассматривается университетская система подготовки профессиональных специалистов, состоящая из студентов, преподавателей и среды обучения [1]. Если учесть, что образовательный процесс невозможно описать математически точно из-за неоднородности, многопараметричности, интеллектуальности и сложности взаимодействия элементов структуры, то очень продуктивным способом, которым можно воспользоваться для исследования такого объекта – моделирование на основе искусственных нейронных сетей [2]. Для построения модели неоднородного объекта целесообразно применить нейросетевую методологию как универсальное средство аппроксимации функции, зависящей

как от числовых, так и от качественных данных.

Поэтому целью данной работы является разработка нейросетевой модели, способной функционально описать зависимость получаемых студентом профессиональных знаний и умений от факторов, влияющих на полноту этих знаний, в частности от психофизиологических характеристик студента. Для достижения этой цели необходимо решить следующие задачи:

- определить внешние и внутренние факторы, влияющие на качество усваивания знаний и навыков;
- разработать методику определения ментальных и психофизиологических особенностей преподавателей и студентов;
- разработать структуру и способ обучения нейросетевой модели.

Назначение данной модели состоит в прогнозировании качества профессионального обучения студентов в зависимости от их личностных характеристик и знаний.

Личностные характеристики определяются путём тестирования студента и анализе полученных результатов. Оценки для построения обучающего множества для первой нейросети берутся из экзаменационной ведомости дисциплины.

Остаточные знания студентов по дисциплине определяются во второй нейросети.

Входными данными для этой нейросети выступают: прогнозируемые оценки с выхода первой нейросети, а перечень знаний и умений берутся из учебной программы дисциплины. Обучающее множество для второй нейросети формирует преподаватель, используя свой профессиональный опыт. получаемых студентом профессиональных знаний и умений по одной дисциплине от факторов, влияющих на полноту этих знаний[3].

Структура нейросетевой модели изображена на рис. 1.

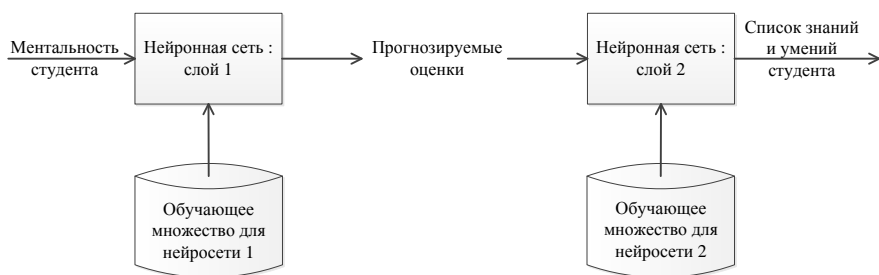


Рисунок - 1. Нейросетевая модель прогноза знаний и умений студента по отдельной дисциплине на основе его ментальности

Также были применены оценочные методики определения ментальных и психофизиологических особенностей студента. Эти методики в совокупности образуют систему, которая определяет ментальный портрет студента. В табл. 1 представлена данная систематизация.

После прохождения всех тестов будет определён ментальный портрет студента, который можно будет использовать при разработке модели передачи знаний[2,4].

Процесс обучения студентов заключается в передаче знаний и навыков от преподавателей. Качество обучения фиксируется в экзаменационной ведомости. Разрабатываемая модель процесса обучения должна формировать на выходе остаточные знания студента по отдельной дисциплине, с которыми он выходит на рынок труда. По ним работодатели решают вопрос о трудоустройстве кандидатов на вакантные должности[5].

Таблица 1. Систематизация характеристик студента

Характеристика ментальности	Способ определения	Оригинальная градация
Уровень интеллекта	Тест на IQ Айзенка	от 0 до 160 баллов
Тип темперамента	Тест Айзенка	4 вида темпераментов
Социальный интеллект	Тест Гилфорда	от 0 до 57 баллов
Уровень мотивации	Тест Гречикова	5 видов

		МОТИВАЦИИ
Уровень креативности	Тест Торренса	от 0 до 70 баллов
Специальные способности	Тест Айзенка по 3 видам способностей	от 0 до 150 баллов за каждый
Умение работать в команде	Самостоятельно	от 0 до 100%
Жилищные условия	Самостоятельно	от 0 до 100%
Состояние здоровья	Самостоятельно	от 0 до 100%
Пол	Самостоятельно	женский или мужской

Прогноз остаточных знаний по одной конкретно взятой дисциплине для одного студента осуществляется в два этапа. На первом этапе прогнозируется экзаменационная оценка, а на втором этапе, исходя из прогнозируемой оценки, формируется усреднённый набор остаточных знаний и умений, соответствующий данной оценке. Первая нейронная сеть будет обучаться на основании ментальных портретов группы студентов и экзаменационной ведомости. Вторая нейросеть – на основании критериев оценки и учебной программы дисциплины, в которой содержится перечень знаний и умений[6].

Структура обеих нейросетей относится к классу однородных многослойных персептронов с полными последовательными связями и с сигмоидальной функцией активации. Обучение нейросетей проводилось по стратегии «обучение с учителем» по алгоритму обратного распространения ошибки. Обучающее множество для второй нейросети составляет преподаватель-профессионал (эксперт) по своей дисциплине, используя утверждённые критерии оценки и учебную программу дисциплины, которая содержит перечень знаний и умений. Для проверки адекватности нейросетевой модели в качестве среды моделирования искусственных нейронных сетей использовался пакет NeuralNetworkToolbox, который входит в стандартную поставку MATLAB[1,6].

Выводы. Предложен подход к нейросетевому моделированию трудно формализуемого процесса профессионального обучения молодых специалистов, основанный на имитации процесса передачи профессиональных навыков и знаний в зависимости от личностных характеристик студентов.

Установлены внешние и внутренние факторы, влияющие на успеваемость студентов и качество усваивания знаний и навыков. Особое внимание было уделено студенту как личности и его месту в процессе обучения. Исходя из этого, была разработана специальная методика, позволяющая анализировать психологические, эмоциональные, природные и физические способности студента.

Первые результаты исследования на NetworkToolbox пакета MATLAB показали, что предложенная идея построения модели может быть положена в основу построения системы моделирования процесса обучения в ВУЗе.

Список литературы

1. Федяев О.И. Нейросетевая модель процесса профессионального обучения молодых специалистов. – В сб. 5-й междунар. науч.-техн. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2015) (Минск, 19-21 февраля 2015 г.). – Минск: БГУИР, 2015. – С. 357-364.

2. Бусленко Н.П. Моделирование сложных систем. – М.: Наука, 1978. – 250 с.

3. Аверченков В.И. и др. Основы математического моделирования технических систем, 2-е изд. – М.: ФЛИНТА, 2011. – 271 с.

4. Шенон Р. Имитационное моделирование систем – искусство и наука: Пер. с англ. – М.: Мир, 1978. – 320 с.

5. Ильин Е.П. Психология творчества, креативности, одарённости. – СПб.: Питер, 2004. – 537 с.

6. Айзенк Г. Новые тесты IQ - М.: Изд-во «ЭСКМО», 2003. – 189.



**WEB-ТЕХНОЛОГИИ И
КОМПЬЮТЕРНЫЙ ДИЗАЙН**

СЕКЦИЯ 4

УДК 004

Е.С. Калинина, А.В. Чернышова

Донецкий национальный технический университет

Кафедра прикладной математики и информатики

E-mail: lena00777@gmail.com, alla@pmi.dgtu.donetsk.ua

АНАЛИЗ ПРОТОКОЛА БЕЗОПАСНОСТИ KERBEROS

Аннотация

Калинина Е.С., Чернышова А.В., Анализ протокола безопасности Kerberos. В тексте данной статьи описан принцип работы протокола Kerberos и возможность его использования в корпоративных системах и сетях. Служба Kerberos, работающая в сети, действует как доверенный посредник, обеспечивая безопасную сетевую проверку подлинности, которая позволяет пользователю работать на нескольких машинах. При общении с каждым объектом сети Kerberos использует сгенерированный общий секретный ключ. На сегодняшний день протокол Kerberos самый безопасный механизм аутентификации, поддерживаемый AD.

Актуальность

На сегодняшний день универсальные системы аутентификации получили широкое распространение в информационно-вычислительных сетях. Причиной этому послужил активный рост требований к защищённости информационно-телекоммуникационных систем. В течение длительного времени основными проблемами были организация масштабируемости системы аутентификации и организация кроссистемного взаимодействия. Однако сейчас существует достаточное количество протоколов, которые решают подобного рода задачи.

Протокол Kerberos был специально разработан для того, чтобы обеспечить надежную аутентификацию пользователей, и

был создан более десяти лет назад в Массачусетском технологическом институте в рамках проекта «Афина». Kerberos – единственный протокол аутентификации Windows, который обеспечивает ограниченное делегирование. [1]

Основная концепция протокола Kerberos

Основу Kerberos составляет протокол аутентификации и распределения ключей Нидхэма-Шредера с третьей доверенной стороной. В протоколе Kerberos участвуют две взаимодействующие стороны и доверенный сервер, который выполняет роль центра распределения ключей. Этот протокол успешно работает при условии, что часы каждого участника синхронизированы с часами сервера. Система Kerberos имеет структуру типа клиент-сервер и состоит из клиентских частей и сервера Kerberos. Клиентами могут быть пользователи, а также различного рода независимые программы, выполняющие такие действия, как загрузка удаленных файлов, отправка сообщений, доступ к БД, доступ к принтерам, получение привилегий у администратора и т.п. Сервер Kerberos KS делится на две части: сервер аутентификации AS (Authentication Server) и сервер службы выдачи мандатов TGS (Ticket Granting Service), которые физически могут быть совмещены. Информационными ресурсами управляет сервер RS. Секретные службы, которые требуют проверку подлинности, и клиенты, которые хотят использовать эти службы, регистрируют в Kerberos свои секретные ключи. Kerberos хранит БД о клиентах и их секретных ключах, наличие которых позволяет создавать зашифрованные сообщения, которые направляются клиенту или серверу. Успешное расшифрование этих сообщений является гарантией прохождения аутентификации всеми участниками протокола. Также Kerberos создаёт сеансовые ключи (session key), которые выдаются клиенту и серверу. Этот ключ используется для шифрования сообщений и уничтожается после окончания сеанса. [2]

Работа системы Kerberos

Процесс идентификации и аутентификации пользователя в системе Kerberos 5 можно описать следующим образом (рис.1) [2]:

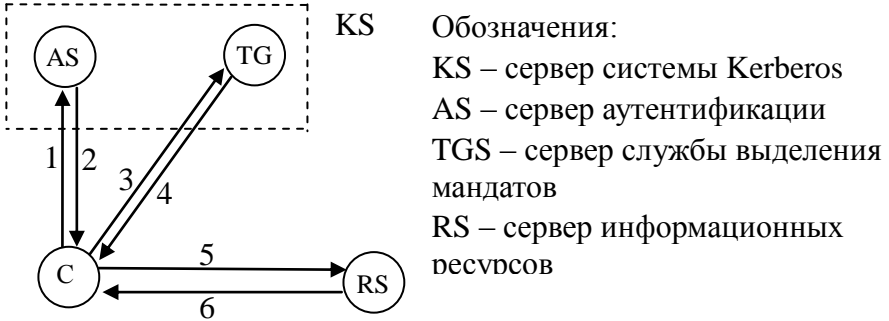


Рис.1. Процесс идентификации и аутентификации пользователя в системе Kerberos 5.

Основные шаги работы Kerberos:

1. Клиент вводит имя пользователя, пароль и домен для получения доступа к ресурсу сети и отправляет запрос серверу аутентификации (AS). Причём, имя клиента передаётся в открытом виде, а текущее время на рабочей станции клиента передаётся в зашифрованном виде и является аутентификатором.

2. Сервер AS проверяет, есть ли такой клиент, выявляет мастер ключ клиента, который основан на пароле и расшифровывает аутентификатор (получает время отправки запроса). Разница текущего времени и времени отправки запроса от клиента не должно превышать определённого значения, которое указано в политике протокола Kerberos. После этого, Сервер AS высылает клиенту билет на получение билета (TGT), формирует ключ сессии (обеспечивает шифрование данных при обмене между AS и клиентом), идентификатор на доступ к

серверу службы выделения мандатов TGS (Ticket-Granting Service) и время жизни билета.

3. Клиент получает данные от сервера AS, расшифровывает свою часть для получения сессионного ключа Клиент/TGS и отправляет запрос TGS мандат на обращение к серверу AS. Запрос содержит полученный ранее TGT, идентификатор сервиса и аутентификатор, зашифрованный на сессионном ключе Клиент/TGS.

4. TGS получает и извлекает TGT, расшифровывает его с использованием секретного ключа TGS. В результате TGS получает сессионный ключ Клиент/TGS, которым расшифровывает аутентификатор. После этого он генерирует сессионный ключ клиент/сервис и посылает ответ клиенту. Ответ содержит:

- билет сервиса, который содержит ID клиента, его сетевой адрес, время действия билета и сессионный ключ клиент/сервис, зашифрованные секретным ключом сервиса.

- идентификатор сервиса, сессионный ключ клиент/сервис и время жизни билета, зашифрованные на сессионном ключе клиент/TGS.[3]

5. Клиент получает ответ и соединяется с сервером RS, посылая ему зашифрованный ранее билет сервиса, новый аутентификатор, зашифрованный на сессионном ключе клиент/сервис, включающий метку времени и ID клиента.

6. Сервер RS получает данные от клиента, расшифровывает билет, используя свой секретный ключ, и получает сессионный ключ клиент/сервис. Этим новым ключом он расшифровывает аутентификатор и посылает сообщение о готовности обслужить клиента. Метка времени, указанная клиентом, обновляется (метка + 1) и высылается клиенту. Клиент проверяет корректность обновления метки, и если всё верно, то клиент начинает посылать запросы на сервер. [4]

Приведенная модель может полноценно функционировать в том случае, если обеспечена полная целостность и конфиденциальность передаваемой информации. Без строгого обеспечения информационной безопасности клиент С не может

отправлять серверам AS, TGS и RS свои запросы и получать доступ к обслуживанию сети.

Использование протокола Kerberos в корпоративных системах и сетях

В настоящее время множество ОС поддерживают данный протокол, в число которых входят:

- ОС семейства Windows, которые используют Kerberos как метод аутентификации в домене между участниками.
- UNIX и UNIX подобные ОС (Apple Mac OS X, Red Hat Enterprise Linux 4, FreeBSD, Solaris, AIX, OpenVMS). [4]

Системы аутентификации, которые базируются или реализуются на основе протокола Kerberos, являются распространёнными, но большая часть из них – дорогие закрытые программные продукты. Наиболее важные реализации протокола:

- KTH Kerberos;
- Heimdal Kerberos;
- MIT Kerberos;
- GNU Shishi;
- Microsoft Kerberos;
- Реализация в составе Apple Mac OS X;
- Реализация в составе Red Hat Linux 4;
- Solaris Kerberos Service;
- FreeBSD.
-

Достоинства и недостатки работы протокола

В реализации протокола Kerberos существуют потенциально уязвимые места:

- возможно кеширование и повторное использование старых удостоверений. Хотя метки должны предотвратить такую возможность, удостоверения могут использоваться повторно в течение времени жизни мандата. Использование удостоверений основано на том, что все часы сети синхронизированы. В случае

если время компьютера будет установлено неправильно, то старое удостоверение может снова быть использовано. Большинство сетевых протоколов поддержки единого времени небезопасны, поэтому такая возможность несёт в себе серьёзную опасность;

- чувствительность к вскрытиям с угадыванием пароля. Злоумышленник может записать мандаты, а затем попытаться их расшифровать. В большинстве случаев пользователь редко выбирает хороший пароль, поэтому, если у злоумышленника будет достаточное количество мандатов, у него появятся все шансы на то, чтобы раскрыть пароль. [5]

К достоинствам протокола можно отнести:

- эффективный доступ к ресурсам и проверка подлинности, которые заключаются в том, что не требуется постоянный переход от сервера с ресурсами к контроллеру домена. Клиент получает «билет», который использует на протяжении всего сеанса или до истечения его срока, для получения доступа к ресурсам; [5]

- эффективное взаимодействие с доменами в лесу. Так как протокол Kerberos основан на спецификациях отслеживания, то все доверительные связи конфигурируются автоматически и являются двухсторонними и поддерживаются между всеми доменами данного леса;

- пользователь вводит пароль один раз за сессию, что позволяет ему иметь доступ ко всем сервисам без повторного ввода пароля. Данное свойство называется Single Sign-On;

- высокий уровень безопасности. При любых взаимодействиях не осуществляется передача паролей, значений хеша паролей в открытом виде.

Выводы

Таким образом, протокол проверки подлинности Kerberos отличается гибкостью и эффективностью использования, а также гарантирует повышенный уровень безопасности. С активным

развитием интернета, электронной коммерции и виртуальных частных сетей, этот протокол является одним из тех, которые удовлетворяют всем необходимым требованиям безопасности на сегодняшний день. Также протокол может использоваться в сочетании с различными криптографическими схемами. В настоящее время ведутся работы над улучшением протокола Kerberos, включая модернизацию управления ключами с помощью криптографии с открытыми ключами и интерфейса интеллектуальных карточек.

Список литературы

1. Технология Kerberos для обеспечения безопасности MOSS 2007 [электронный ресурс]. – Режим доступа: <http://www.osp.ru/win2000/2008/05/5528562/>
2. Протокол Kerberos [электронный ресурс]. – Режим доступа: <http://ypn.ru/427/kerberos-protocol/>
3. Kerberos [электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Kerberos>
4. Принципы аутентификации по протоколу Kerberos [электронный ресурс]. – Режим доступа: <http://itband.ru/2010/12/kerberos1/>
5. Брюс шнайер. Прикладная криптография 2-е издание. Протоколы, алгоритмы и исходные тексты на языке С. – 1994 – 816 с.

УДК 004

М.В. Клокова, А.В. Чернышова

Донецкий национальный технический университет

Кафедра прикладной математики и информатики

E-mail: marizombie@outlook.com, alla@pmi.dgtu.donetsk.ua

АНАЛИЗ СРЕДСТВ ЗАЩИТЫ ИНФОРМАЦИИ В КОРПОРАТИВНЫХ СЕТЯХ

Аннотация

Клокова М.В., Чернышова А.В. Анализ средств защиты информации в корпоративных сетях. В данной статье проведен анализ современных методов и способов защиты информации в корпоративных сетях. Рассмотрены криптографические средства, основные алгоритмы шифрования данных, технологии защиты от проникновений в сеть, а также составлен список требований к системам комплексной защиты информации для поддержания высокого уровня безопасности как внутри сети, так и при отправке данных за ее пределы.

Актуальность

“Жить – значит быть источником и приемником информации” (А. Конопатский). Информация в современном обществе – одна из самых ценных вещей в жизни, требующая защиты от несанкционированного проникновения лиц, не имеющих к ней доступа. За последние несколько лет значительно возрос интерес частных и государственных компаний на рынке средств защиты информации.

Изложение материалов исследования

Для защиты информации применяются различные меры и способы, начиная с организационно-режимных и кончая применением сложных программно-аппаратных комплексов.

Одним из путей решения проблемы защиты информации, а точнее - решения небольшой части вопросов из всего спектра мер защиты, является криптографическое преобразование информации.

Криптографическими средствами защиты называются специальные средства и методы преобразования информации, в результате которых маскируется ее содержание. Основными видами криптографического закрытия являются шифрование и кодирование защищаемых данных. Рассмотрим подробнее саму суть шифрования.

Шифрование – обратимое преобразование информации в целях сокрытия от неавторизованных лиц, с предоставлением, в это же время, авторизованным пользователям доступа к ней. Этим занимается наука о методах обеспечения конфиденциальности – криптография. Задача криптографии, т.е. тайной передачи, возникает только для информации, которая нуждается в защите. В таких случаях считается, что информация содержит тайну или является защищаемой, приватной, конфиденциальной, секретной. [1]

Существует множество разных криптографических алгоритмов. Назначение этих алгоритмов — защита информации. Защищать же информацию приходится от разных угроз и разными способами. Чтобы обеспечить надежную и адекватную защиту с помощью криптоалгоритма (КА), нужно понимать, какие бывают КА и какой тип алгоритма лучше приспособлен для решения конкретной задачи.

Алгоритм симметричного шифрования требует наличия одного ключа для шифрования и дешифрования сообщений. Такой ключ называется общим секретным, поскольку все пользователи, участвующие в обмене данными, имеют один и тот же ключ.

Алгоритм асимметричного шифрования требует использовать один ключ для шифрования данных и другой, но взаимосвязанный с ним ключ — для дешифрования. Один из ключей в такой схеме доступен любому, кто его запрашивает.

Такой ключ называется открытым. Другой ключ известен только владельцу и называется секретным.

Внедрение криптосредств в системы защиты персональных данных

На сегодняшний день деятельность большинства крупных компаний связана с обработкой персональных данных различного типа, к защите которых выдвигается ряд требований. Прежде чем их выполнить, руководству компании необходимо построить модель угроз персональным данным и на ее основе разработать систему защиты, в состав которой должно входить средство криптографической защиты информации (СКЗИ).

Внедренное в систему защиты персональных данных СКЗИ должно соответствовать следующим требованиям:

- криптографическое средство должно штатно функционировать совместно с другими техническими и программными средствами;
- при обработке персональных данных для обеспечения их безопасности должны использоваться сертифицированные криптосредства.

По уровням защиты криптографические средства делят на 6 классов: КС1, КС2, КС3, КВ1, КВ2, КА1. Выбор определенного класса средств напрямую зависит от субъекта атаки (нарушителя), категория которого становится известна благодаря оператору в модели угроз. [2]

Средства криптографической защиты сегодня эффективно используются компаниями и организациями для защиты персональных данных и являются одной из важнейших составляющих в системах защиты персональных данных.

Защита корпоративной информации

Применительно к компаниям шифрование обычно используется в трех случаях, а именно в целях обеспечения безопасности хранения данных, защиты информации при

передаче через открытые каналы связи и по локальной сети и, наконец, для цифровых подписей. Все представленные задачи решаются разными средствами с применением различных криптографических технологий.

Итак, еще одно применение СКЗИ – это защита конфиденциальной информации компании. В сфере компьютерной безопасности для защиты корпоративной информации используется два принципиально разных подхода к защите от проникновений в сеть.

Первый и более старый из них это IDS (Intrusion Detection Systems). IDS – это система призванная обнаружить попытки проникновения в частную сеть и сообщить системному администратору о факте вторжения. Эта технология защиты информации используется довольно давно и уже завоевала популярность среди заказчиков.

Однако, многие аналитики считают, что сегодня существует более эффективный и удобный способ борьбы с хакерами. Эта система – Intrusion Prevention System, IPS, которая служит для предотвращения нападений. Под ней подразумевается набор технологий, которые появились на стыке межсетевых экранов и систем обнаружения нападений IDS. От межсетевых экранов в IPS взят принцип активного вмешательства в сетевое взаимодействие или поведение программ, а от IDS – интеллектуальные методы мониторинга происходящих событий. Таким образом, IPS не только обнаруживает нападения, но и пытается предотвратить их.

Среди продуктов IPS выделяют пять составляющих, каждая из которых может как работать самостоятельно, так и комбинироваться с другими. Это сетевая IDS, коммутаторы седьмого уровня, экран приложений, гибридные коммутаторы и ловушки.

Следует отметить, что IPS разных типов хорошо интегрируются в достаточно интеллектуальную систему защиты, в которой каждый элемент дополняет другие. При этом они не конкурируют с уже существующими средствами

информационной безопасности: межсетевыми экранами, IDS, антивирусами и др., поскольку дополняют их. [3]

Отдельное внимание стоит уделить технологии виртуальных защищенных сетей (Virtual Private Network). VPN способна обеспечить конфиденциальность, целостность данных и имитостойкость на уровне IP-пакетов. На основе этой технологии обеспечивается защищенное соединение между подсетями и компьютерами.

Технология VPN может работать с множественными алгоритмами шифрования и сложными конфигурациями тоннелей и защищенных периметров. Криптографическая стойкость обеспечивается применением соответствующих криптографических алгоритмов (3DES, AES, ГОСТ).

Главное преимущество данной технологии – прозрачность работы системы защиты: приложения работают так же свободно, как и в открытой сети, ничего не подозревая о работе VPN. [4]

Утечка информации

Согласно последним исследованиям, для многих IT компаний в последнее время стали проблемой инсайдеры – сотрудники, которые либо намеренно, либо случайно становятся источниками утечки информации. По мнению А. Крячкова, директора по продуктам компании Aladdin - ведущего разработчика и поставщика средств аутентификации, продуктов и решений для обеспечения информационной безопасности и защиты конфиденциальных данных в России – рекомендации по защите весьма просты. По его словам необходимо лишь следовать концепции 3А (Аутентификация, Авторизация, Аудит). Данная концепция основывается на следующих требованиях:

- доступ к критичным ресурсам компании должен быть персонифицированным, что позволяет снизить риск отказа пользователей от совершенных ими действий;
- необходимо применять шифрование критически важных данных, представляющих сферу особого интереса инсайдера;

- ключевая информация и другие секретные данные пользователя должны храниться на личном съемном носителе (например, на смарт-карте или USB-ключе), что повысит личную ответственность каждого сотрудника;
- в качестве профилактики и для снижения риска утечки информации важно проводить регулярный мониторинг всех действий пользователей в сети.

А.П. Кекишев, главный инженер ЗАО “Монлайн” – компании, которая специализируется на создании современных информационно-вычислительных систем, систем безопасности – считает, что определяющим фактором при выборе средств защиты информации является ее стоимость. Поскольку неразумно тратить на защиту средства большие, чем ее предполагаемая цена. Разумными считаются расходы в размере не более 10% стоимости информации. [5]

Информацию нужно защищать не только от несанкционированного доступа, но и от уничтожения. С этой целью в любой компании действует система резервного копирования данных. Специалисты в области информационной безопасности рекомендуют дополнять системы резервного копирования криптографическими модулями для защиты данных, размещаемых на внешних жестких дисках.

Выводы

По итогам данной статьи можно сказать, что криптографические методы защиты являются одним из лучших вариантов сокрытия информации на сегодняшний день.

При планировании защиты необходимо знать, кого и какая именно информация будет интересовать, какова ценность этой информации и каковы уязвимости системы хранения.

Система защиты должна быть комплексной. Стоит обеспечивать защиту сразу на трех уровнях: программном, программно-аппаратном и аппаратном.

В то же время система защиты информации должна быть гибкой и адаптированной. В этом главную роль играет

администрирование: регулярная смена паролей и ключей, строгий порядок их хранения, анализ журналов регистрации событий в системе, распределение полномочий пользователей в системе и другое.

Для шифрования информации использовать симметричные алгоритмы шифрования, для шифрования ключей использовать ассиметричные алгоритмы.

Для контроля целостности пользовательских данных использовать электронно-цифровую подпись.

Предусмотреть возможность реализации резервного копирования информации стандартными средствами операционных систем.

Система должна иметь простой и понятный интерфейс.

Список литературы

1. Саймон Сингх. Книга шифров. Тайная история шифров и их расшифровки. — АСТ, Астрель, 2007 г. — 446 с.

2. Использование шифрования в компаниях России: [электронный ресурс]. Режим доступа: <http://habrahabr.ru/company/cybersafe/blog/220023/>

3. Современная криптография: [электронный ресурс]. Режим доступа: <http://www.osp.ru/cio/2006/08/2681235/>

4. Стандарты сетевой защиты информации: [электронный ресурс]. Режим доступа: <http://www.s-terra.com/solutions/standards/>

5. Исследование “Средства защиты информации”: [электронный ресурс]. Режим доступа: http://www.itsec.ru/articles2/Oborandteh/issledovanie_sredstva_zashit_y

УДК 004.942

И.А. Офицеров, К.А. Пшеничный

Санкт-Петербургский национальный исследовательский
университет информационных технологий, механики и оптики
Санкт-Петербург

E-mail bulldozer.spb@gmail.com

**ИССЛЕДОВАНИЕ ПРИМЕНИМОСТИ МЕТОДА
ДИАГРАММ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ ЗАЩИТЫ
ОТ УГРОЗ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРИ
ПЕРЕДАЧЕ ДАННЫХ ПО ОПТОВОЛОКОННЫМ
КАНАЛАМ**

Аннотация

И.А. Офицеров, К.А. Пшеничный. Исследование применимости метода диаграмм последовательностей для защиты от угроз информационной безопасности при передаче данных по оптоволоконным каналам. Передача данных по оптоволоконным каналам в настоящее время является одним из основных видов информационного обмена. Вместе с тем, обеспечить контроль сохранности столь разветвленных и протяженных объектов не всегда представляется возможным. При их повреждении неизбежно возникают угрозы информационной безопасности пользователям этих каналов.

Стоит отметить, что физическая целостность кабеля может быть нарушена (умышленно или случайно) в результате множества причин – например, разрывов снарядов в ходе боевых действий, землетрясения или даже падения метеорита. При этом важно отметить, что данный список заведомо неполон, поскольку невозможно учесть и классифицировать все возможные воздействия внешней среды, включая человеческий фактор. В то же время, было бы крайне желательно и полезно каким-то образом предвидеть все мыслимые угрозы, чтобы разработать универсальные рекомендации по защите от них.

Решение данной проблемы видится в формальном представлении рассматриваемого технического объекта (в данном случае, системы оптоволоконной связи) и возможных угроз его функционированию в виде модели, называемой диаграммой последовательностей.

Введение

Ценность информации зависит не только от её содержания, но и от своевременности её доставки получателю. Несвоевременность доставки информации может не только снизить её стоимость до нуля, но и нанести немалый ущерб. В связи с этим, обеспечение беспрепятственной передачи информации является очень важным аспектом информационной безопасности (далее – ИБ).

Часто причинами сбоев в передаче информации являются такие непреднамеренные угрозы, как обрыв линий связи при прокладке новых инженерных коммуникаций или при ремонте уже существующих^[1]. Поэтому важно найти такой метод, который позволил бы увидеть различные возможные сценарии и выбрать определенный план действий при наступлении этой угрозы.

Решение данной проблемы видится в формальном представлении рассматриваемого технического объекта (в данном случае, системы оптоволоконной связи) и возможных угроз его функционированию в виде модели, называемой диаграммой последовательностей^[2].

Насколько известно автору, диаграммы последовательностей никогда ранее не применялись для изучения угроз информационной безопасности.

Постановка задачи

Передача данных по оптоволоконным каналам в настоящее время является одним из основных видов информационного обмена^[3]. Вместе с тем, обеспечить контроль сохранности столь разветвленных и протяженных объектов не всегда представляется возможным. При их повреждении неизбежно возникают угрозы

информационной безопасности пользователям этих каналов. Так, в ходе прокладки инженерных коммуникаций в посёлке Сертолово Ленинградской области РФ 8 июля 2008 года был неумышленно оборван магистральный волоконно-оптический кабель (по специальной терминологии – транк), принадлежавший провайдерам «Северо-Западный Телеком» и ООО «Икс-Трим». В результате этого без услуг связи остались абоненты этих компаний. Похожая ситуация произошла между городами Бежецк и Удомля Тверской области РФ 29 января 2009 года, когда лишились связи абоненты таких операторов связи, как RETN.net и «Билайн-Бизнес». Не пострадали клиенты тех провайдеров, которые перенаправили трафик на другие каналы, однако абоненты ряда домашних сетей, не имеющих резерва, так и остались без доступа в сеть Интернет^[4].

Стоит отметить, что физическая целостность кабеля может быть нарушена (умышленно или случайно) не только при проведении земляных работ, но и в результате множества иных причин – например, разрывов снарядов в ходе боевых действий, землетрясения или даже падения метеорита. При этом важно, что данный список заведомо неполон, поскольку невозможно учесть и классифицировать все возможные воздействия внешней среды, включая человеческий фактор. В то же время, было бы крайне желательно и полезно каким-то образом предвидеть все мыслимые угрозы, чтобы разработать универсальные рекомендации по защите от них.

Моделирование ситуации

Для решения данной задачи необходимо, прежде всего, прибегнуть к словесному описанию ситуации, а затем визуализировать её. Визуализация рассмотренной ранее ситуации с обрывом транка будет проводиться путем построения модели с использованием метода диаграмм последовательностей. В качестве элементной базы модели будут использоваться графические элементы нотации моделирования бизнес-процессов (далее – BPMN)^[5].

Реализация угрозы происходит следующим образом.

Экскаваторщик проводит земельные работы. Если зона проведения работ не совпадает с зоной прокладки транка, то транк не может быть поврежден в результате их проведения, следовательно, не может быть нарушена и доступность информации. Если же зона проведения работ совпадает с зоной прокладки транка, то существует вероятность того, что транк будет поврежден в результате их проведения. Например, невнимательность и спешка экскаваторщика могут привести к тому, что он случайно заденет транк ковшом экскаватора, в результате чего произойдет обрыв транка. В этом случае передача информации по транку прерывается. Таким образом, реализовывается угроза доступности информации для пользователей этого транка.

При выполнении работ в непосредственной близости от прокладки транка может быть осуществлен дополнительный контроль за их проведением. Это позволяет вовремя обнаружить пролегающий транк и не допустить его повреждение, и, как следствие, предотвратить реализацию угрозы доступности информации.

Если такой контроль не проводится, и транк повреждается, то доступность информации нарушается. Существует два пути восстановления доступности информации:

1. восстановить поврежденный транк;
2. подключиться к резервному каналу передачи информации.

Реализация первого способа требует гораздо больше времени, чем второго. Так как в течение всего этого времени информация будет оставаться недоступной, уровень негативного влияния последствия реализации угрозы в первом случае будет гораздо выше, чем во втором.

Таким образом, дальнейшее развитие ситуации возможно двумя путями:

1. если пользователь имеет доступ к резервному транку, то он, подключившись к нему, возобновляет доступность информации;
2. если у пользователя нет резервного транка, то

информация для данного пользователя остается недоступной.

При отсутствии у пользователя резервного транка остается единственный способ возобновления её доступности – отремонтировать поврежденный транк. Для этого, прежде всего, необходимо сообщить о проблеме специальной организации, занимающейся ремонтом волоконно-оптических линий связи (ВОЛС), указав в сообщении адрес поврежденного канала связи.

Организация по ремонту ВОЛС, получив сообщение о повреждении транка, прибывает по указанному адресу, проводит поиск места повреждения транка и выполняет ремонт поврежденного участка.

После проведения ремонта поврежденного участка транка доступность информации для пользователей данного транка возобновляется.

Основываясь на приведенном выше словесном описании ситуации, методе построения диаграмм последовательностей и нотации BPM, вербальная модель рассмотренной ситуации с обрывом транка и соответствующая ей субъектно-предикатная модель^[6] принимают вид, показанный в приложениях 1 и 2, соответственно.

Данная диаграмма позволяет сделать вывод о том, что предотвратить подобную угрозу практически невозможно в силу случайности характера её возникновения. Однако, можно существенно снизить негативные последствия её реализации при наличии резервного канала передачи информации и своевременного подключения к нему «пострадавшего» клиента. Кроме того, при проведении земляных работ на участках, где вероятность повреждения транка при неосторожных действиях достаточно высокая, целесообразно осуществлять дополнительный контроль за их выполнением. Это позволит вовремя обнаружить пролегающий транк и не допустить его повреждение, и, как следствие, предотвратить реализацию угрозы доступности информации.

Заключение

Результаты моделирования показывают, что диаграмма

последовательностей весьма удобна для визуализации процессов. Благодаря небольшому и, вместе с тем, универсальному набору графических элементов можно легко визуализировать самые различные ситуации. Кроме того, правила применения данных элементов просты, что позволяет быстро и без особых усилий научиться строить подобные диаграммы. Диаграмма легко читается и воспринимается, а её построение не требует много времени.

На построенной модели, словно на карте, можно выделять наиболее уязвимые места, в которых возможно возникновение потенциальных угроз информационной безопасности, определять алгоритм «избегания» возникшего негативного сценария, обеспечив, таким образом, безопасное функционирование технического объекта. Это позволяет не только минимизировать негативный эффект от уже возникшей угрозы, но и вовсе предотвратить её возникновение.

Кроме того, начальное событие может быть инициировано не только, как в рассмотренной ситуации, экскаваторщиком, но и, например, разрывом снаряда, падением метеорита или воздействием на транк иных факторов, однако модель процесса, при этом, останется прежней.

Таким образом, данная модель позволяет формально представить рассмотренную информационную угрозу и применима к множеству других воздействий на транк вплоть до боевых действий или падения космических тел, если только они могут быть проявлены на территории размещения объекта.

Представляется вполне возможным, что при многократном построении диаграмм последовательностей для моделирования сценариев информационных угроз удастся выявить схожие последовательности событий или действий и, таким образом, выделить формальные типы угроз независимо от того, в каких сферах человеческой деятельности они проявляются. Далее, внутри каждого формального типа можно анализировать опыт из разных сфер деятельности, будь то логистика или банковская сфера, и с требуемым обобщением обмениваться этим опытом. Дальнейшие исследования в данном направлении позволят

подойти к формальной классификации информационных угроз и способов защиты от них.

Список литературы

1. SPBITRU. Разрыв магистральных линий - часть 4. Москва и Питер потеряли связь. [электронный ресурс]. – режим доступа: <http://spbit.ru/news/n56359/>, свободный [12.05.2015].

2. Wikipedia. Sequence diagram [электронный ресурс]. – режим доступа: https://en.wikipedia.org/wiki/Sequence_diagram, свободный [12.05.2015].

3. Джонсон Г., Грэхэм М. Высокоскоростная передача цифровых данных. – М.: Вильямс, 2005. – С. 136-139.

4. SPBITRU. Газификация на пути интернетизации [электронный ресурс]. – режим доступа: <http://spbit.ru/news/n50981/>, свободный [21.03.2015].

5. Нотация BPM. Элементы нотации BPM. [электронный ресурс]. – режим доступа: http://www.elma-bpm.ru/bpmn2/7_2.html#7_2, свободный [21.03.2015].

6. Pshenichny, C.A., and Mouromtsev, D.I., 2015. Grammar of Dynamic Knowledge for Collaborative Knowledge Engineering and Representation. In: Diviacco, P., Fox, P., Pshenichny, C., Leadbetter, A. (Eds): Collaborative Knowledge in Scientific Research Networks, IGI Global, pp 323-354.

УДК 004.056.55

Д.С. Сипаков, Н.Е. Губенко

Донецкий национальный технический университет, г. Донецк
кафедра компьютерных систем мониторинга

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ЛИНГВИСТИЧЕСКОЙ СТЕГАНОГРАФИИ

Аннотация

Сипаков Д.С., Губенко Н.Е. Сравнительный анализ методов лингвистической стеганографии. Были проведены исследования методов лингвистической стеганографии, с целью определения достоинств и недостатков, а так же для определения наиболее качественного метода.

***Ключевые слова:** стеганография, лингвистика, защита данных.*

Постановка проблемы.

Развитие информационных технологий и IT индустрии в целом повлекло за собой возникновения ряда проблем связанных с предоставлением и обработкой информации. Одной из самых актуальных на сегодняшний день является проблема защиты информации. Для обеспечения защиты информации при передаче ее между сотрудниками или отделами используются стеганографические алгоритмы. Сейчас популярность набирает так называемая лингвистическая стеганография — скрывает сообщение, изменяя параметры элементов, из которых состоит носитель или, создавая новый носитель таким образом, чтобы он содержал передаваемое сообщение. Другими словами, целью лингвистической стеганографии является скрытие факта передачи информации, а также ее шифрование.

Цель статьи.

Выполнить сравнительный анализ методов лингвистической стеганографии, выявить достоинства и недостатки исследуемых методов.

Решение задач и результаты исследований.

Для проведения сравнительного анализа были выбраны следующие методы лингвистической стеганографии: семаграмма, фонетика, жаргон, скрытое кодирование. Для каждого метода были установлены достоинства и недостатки.

1 Семаграмма

Семаграмма – скрытие информации с помощью знаков или символов. Например, знак рукой, головой, определенная последовательность чисел или слов, все это семаграммы. Такие знаки не бросаются в глаза, и не заметны для стороннего человека. Иногда использование визуальных семаграмм – единственный способ связи с друзьями и коллегами.



Рисунок 1 — Схема семаграммы

Текстовые семаграммы – это послания, скрытые внутри текста. Заглавные буквы, подчеркивания, особенности почерка, пробелы между буквами и словами – все они могут быть использованы для передачи какого-либо сообщения.

Достоинства:

- незаметно для сторонних лиц;
- простота реализации.

Недостатки:

- из-за простоты реализации, достаточно легко можно получить скрытую информацию.

2 Фонетика

Данный способ можно применить, если точно знать на какой язык запрограммирован фильтр, который "отлавливает" слова на том языке, который преимущественно используют жители страны. Иногда (вдобавок) и на том языке, который сильно распространен или используется на веб-сайтах (английский, французский). Конечно, нельзя с уверенностью сказать, как в

точности запрограммирован фильтр. Но чтобы приблизиться к пониманию, можно использовать фонетически схожие слова. Этот способ наиболее подходит, если вы используете алфавит, отличный от принятого в вашей стране (например, латинский вместо русского)[3].

политзаключенный politzaklyutchenny

Рисунок 2 — Пример фонетики

Достоинства:

– простота реализации.

Недостатки:

- легко получить скрытую информацию;
- фактически информация не скрывается.

3 Жаргон

Использование жаргонов в тексте может озадачить постороннего читателя. Более того жаргонные слова могут быть придуманы членами переписки. Для реализации данного метода создается база данных слов и соответствующих жаргонов, при скрытии слова заменяются. Получатель, должен обязательно знать использованные жаргонизмы что бы понять текст сообщения. Лучше выбирать такие слова, которые сохраняют текст-"носитель" в понятном, читаемом виде, даже если воспринимать жаргонизмы "как есть"[3].

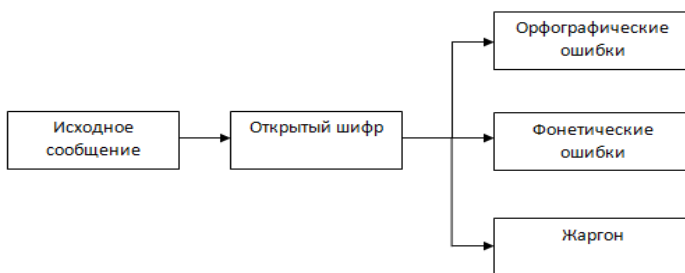


Рисунок 3 — Схема метода жаргонов

Достоинства:

- простота реализации;
- если сохранять текст в читаемом виде, не возникнет подозрений о наличии скрытой информации.

Недостатки:

- перечень слов ограничен участниками переписки;
- получатель должен знать жаргоны;
- при неудачном использовании жаргонов, злоумышленник может догадаться и понять скрытый смысл сообщения.

4 Скрытое кодирование

Частный случай лингвистической стеганографии, наиболее сложен в реализации, однако обеспечивает высокую скрытность информации. В данном методе используется специальная функция, которая шифрует и дешифрует сообщение для передачи. Используется так называемый «контейнер», в который помещается скрываемое сообщение, при получении определенным алгоритмом сообщения достается из «контейнера».

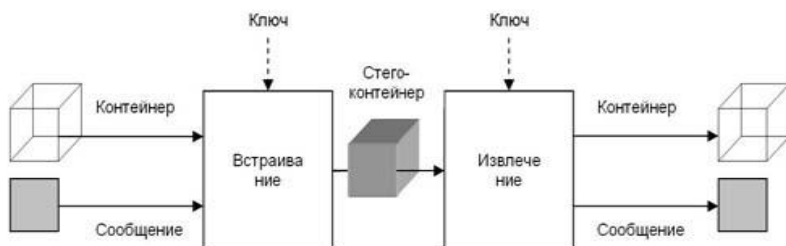


Рисунок 4 — Общая схема передачи скрытого сообщения

Достоинства:

- при наличии программы шифрования дешифрования, получателю не обязательно знать алгоритм скрытия информации;
- крепкая стойкость к дешифрованию;

- тяжело понять, что сообщение-шифр имеет скрытый подтекст.

Недостатки:

- сложность реализации;
- при генерации текста «сообщения-контейнера» тяжело получить читаемый вид передаваемого сообщения.

После проведения анализа исследуемых методов стеганографии, была составлена таблица сравнения.

Таблица 1 — Сравнительная таблица методов стеганографии

Наличие достоинства/недостатка	Метод лингвистической стеганографии			
	Семаграмма	Фонетика	Жаргон	Скрытое кодирование
Сложность реализации	-	-	-	+
Легкость определения наличия зашифрованного сообщения	-	+	-	-
Легкость дешифрования сообщения	+	+	+	-
Искусственные ограничения (например: запас слов у отправителя и получателя, или другие условности между ними)	+	+	+	-

Анализируя таблицу 1, можно сделать вывод, что наиболее качественным методом стеганографии является скрытое кодирование. Данный метод гарантирует полное скрытие самого факта передачи сообщения, не имеет ограничений, однако он же является самым сложным в реализации.

Выводы

Таким образом, при проведении исследований было выяснено, что наиболее стойким методом лингвистической стеганографии является скрытое кодирование. Данный метод имеет сложную реализацию, которая связана с генерированием слов с помощью определенной функции, однако именно данный метод маскирует сам факт передачи скрытого сообщения и получить это сообщение без определенного ключа (функции) практически невозможно. Поэтому целесообразно тратить усилия на реализацию именно этого метода.

Список литературы

1. Life-Prog [Электронный ресурс] – Режим доступа: http://www.life-prog.ru/1_6417_aktualnost-problemi-zashchiti-informatsii.html
2. Википедия [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Стеганография>
3. Digital Security [Электронный ресурс] – Режим доступа: https://equalit.ie/eseaman/russian/chapter2_8.html
4. Википедия [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Жаргон>

УДК 004.942

Р.С. Тимонин, К.А. Пшеничный

Санкт-Петербургский национальный исследовательский
университет информационных технологий, механики и оптики
Санкт-Петербург

E-mail cpshenichny@yandex.ru

ИССЛЕДОВАНИЕ ПРИМЕНИМОСТИ МЕТОДА ДИАГРАММ ДЕЯТЕЛЬНОСТИ В СФЕРЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Аннотация

Тимонин Р.С., Пшеничный К.А. Исследование применимости метода диаграмм деятельности в сфере информационной безопасности. Целью данной работы является исследование применимости метода диаграмм деятельности в сфере информационной безопасности. Результатом работы является модель представления анализируемого процесса на примере банковской деятельности. Такие модели можно применять в различных случаях, связанных с выявлением и анализом нетипичных значений, и в конечном счете прийти к формальной классификации информационных угроз в зависимости от свойств модели.

Введение. Достижения в сфере информационных технологий и массовое внедрение средств автоматизации в различных сферах деятельности, наряду с очевидной пользой, многократно увеличивают разнообразие возможных каналов утечки информации и, соответственно, угроз несанкционированного доступа^[1] (далее – НСД) к ней. Поэтому весьма полезной для обеспечения безопасности автоматизированной системы (далее – АС) представляется разработка наглядной, максимально исчерпывающей модели угроз информационной безопасности^[1] (далее – ИБ) объекта информатизации. Вопрос обеспечения ИБ особенно остро стоит в банковской сфере, где ущерб от информационных угроз может быть исключительно велик^[2].

Необходимость проведения настоящей работы вызвана вескими основаниями, изложенными в п.6 Доктрины ИБ РФ^[3], полагать, что применяемые в настоящее время в отечественных системах обработки банковской информации организационные меры, а также аппаратно-программные средства защиты не могут обеспечить достаточную степень безопасности субъектов, участвующих в процессе информационного взаимодействия, и не способны в необходимой степени противостоять разного рода угрозам с целью доступа к финансовой информации и дезорганизации работы автоматизированных банковских систем.

Существует множество методов для представления угроз объекта информатизации, однако каждый из них имеет свои ограничения, в частности, отсутствие динамической интерпретации процесса взаимодействия с защищаемым объектом. В данной работе угрозы ИБ, возникающие в банковской деятельности при НСД к АС, будут исследованы и наглядно представлены с помощью метода диаграмм деятельности (activity diagrams)^[4]. Метод диаграмм деятельности применяется для моделирования динамических аспектов поведения систем^[5]. Функционирование АС и воздействие на нее представляет собой динамическое состояние системы, поэтому исследование преднамеренных угроз безопасности АС с помощью метода диаграмм деятельности, изучающего динамические аспекты систем, представляет большой практический интерес.

Целью настоящей работы является исследование применимости метода диаграмм деятельности в сфере информационной безопасности. Для достижения данной цели необходимо разработать модель представления процесса, который может быть подвержен преднамеренным угрозам, с учетом всех его аспектов, которые необходимы для данного исследования.

Постановка задачи. В качестве преднамеренной угрозы для примера будет рассмотрена угроза НСД лицом, не имеющего доступа к конкретной АС или отдельным ее элементам, на примере ввода в систему текущих курсов валют. Получив доступ к АС, злоумышленник^[6] может нарушить конфиденциальность,

целостность и доступность обрабатываемой в системе информации, в том числе исказить информацию (ввести ложный курс валюты) в целях получения собственной выгоды.

В автоматизированную систему банка каждый день вводят новые данные по курсам валют. При этом система сравнивает вновь вводимые значения курсов с аналогичными значениями, введёнными вчера. Если разница по модулю не превышает некоторого значения, принятого за предельно допустимую разницу и введённого в систему заранее, вновь вводимое значение принимается системой и может быть использовано при проведении банковских операций. Если превышает, система требует от пользователя подтверждения того, что данное значение является истинным.

Таким образом, если ввести в систему заведомо большее значение предельно допустимой разницы, чем то, которое может иметь место в действительности, и сообщить пользователю фиктивное значение курса, система может не потребовать от пользователя подтверждения при вводе заведомо ложного значения курса, которое, соответственно, будет использовано при проведении операций. Это открывает возможность для мошенничества и является преднамеренной угрозой информационной безопасности банка.

Построение вербальной модели. Теперь можно посмотреть, как эта ситуация может быть представлена с помощью метода диаграмм деятельности (рис. 1).

На рисунке рассматривается вербальная модель процесса (деятельности) установки в АС банка нового курса валют, наглядно отображающая с необходимой точностью последовательность атомарных действий для реализации поставленной задачи (от начала ввода курса и заканчивая его установкой в качестве текущего). Весь практический смысл составления модели заключается в анализе всех этих действий, чтобы проследить, на каком из этапов (действии) может возникнуть потенциальная угроза.

Деятельность, как составной элемент диаграммы, является совокупностью простых действий (в данном случае

последовательных)^[7]. Начало диаграммы исходит из начального состояния^[8]. Далее следует действие «Ввести курс валюты». Это действие влечет за собой ввод нового курса валюты в АС, который устанавливается текущим ежедневно (за исключением выходных и праздничных дней).

Казалось бы, что еще нужно для установки введенного курса валюты текущим курсом. Однако, есть еще несколько простых действий, призванных минимизировать возможные ошибки оператора АРМ в АС при вводе нового курса, а также предотвратить ввод ложного значения курса валюты злоумышленником.

Одним из таких защитных механизмов является следующий далее узел решения. Выполняя проверку на наличие непустого значения в форме ввода курса валюты, он, тем самым, предотвращает возникновение дальнейших ошибок при условии перехода к следующему действию. Естественно, при пустом значении в форме узел решения переведет процесс выполнения деятельности снова к действию «Ввести курс валюты». Этот цикл будет продолжаться до тех пор, пока оператор не введет значение курса, отличное от пустого.

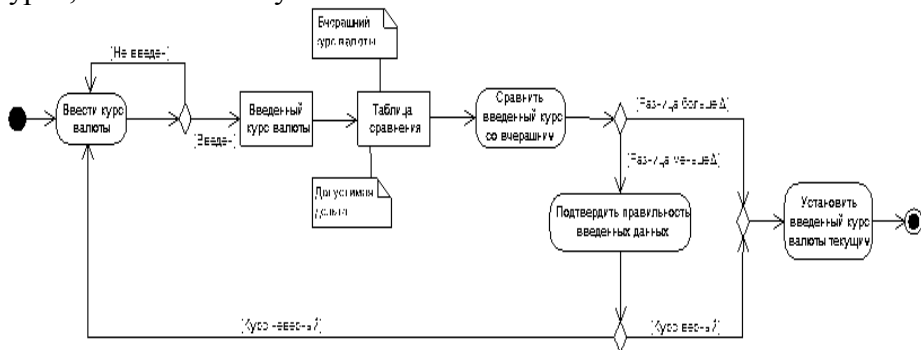


Рис. 1 – Метод диаграмм деятельности. Вербальная модель

За узлом решения в диаграмме следует объект «Введенный курс валют», который в действительности символизирует тот самый курс валют, введенный оператором АРМ.

Введенный курс попадает в таблицу сравнения, в которой помимо вчерашнего курса валют содержится значение, представляющее собой предельно допустимую разницу между вчерашним курсом валюты и только что введенным. То есть по сути это значение является дельтой ввода. Сама таблица сравнения отображается в виде одноименного объекта с прилагающимися к нему комментариями. Каждый из комментариев («Вчерашний курс валюты» и «Допустимая дельта») указывает на нахождение в таблице сравнения этих самых непустых значений.

Теперь, когда в диаграмме инициализированы все необходимые значения для сравнения, состояние деятельности переходит к действию «Сравнить введенный курс со вчерашним». Это действие находит разницу по модулю между значениями введенного курса валюты и вчерашним курсом.

На этом сравнение как таковое не заканчивается. Далее осуществляется переход к очередному узлу решения, который в зависимости от величины значения дельты, полученной в результате сравнения значений курса валют, принимает одно из решений. Либо состояние деятельности переходит к узлу объединения (для узла объединения только при выполнении того или иного действия осуществляется переход к следующему узлу управления) при полученной дельте меньше допустимой дельты, либо в ином случае переходит к еще одному проверочному действию. Следует рассмотреть второй случай. Дельта, полученная при сравнении введенного и вчерашнего курсов валют, оказывается больше допустимой. Здесь можно было бы выдать сообщение об ошибке и предложить заново ввести курс валюты, то есть, по сути, вернуться к начальному состоянию диаграммы и проделать весь путь до этого момента заново. Однако учитывая сложившуюся текущую обстановку с курсом рубля по отношению к таким крупным валютам, как европейская валюта и доллар США, когда курс может измениться в меньшую или большую сторону на несколько единиц или даже десятков значений, этого делать нельзя^{[9][10]}. Допустимая дельта в стабильной экономической ситуации обычно равна десятым

долям, текущая же ситуация непредсказуема^[11]. Поэтому даже если полученная дельта превышает допустимую, процесс не начинается заново с ввода нового курса валюты заново, а переходит далее к подтверждению правильности уже введенного курса. В этой ситуации может играть роль человеческий фактор, поэтому оператор, введивший курс, должен перепроверить его значение. Соответственно, если оператор действительно ошибся, он переходит к начальному состоянию деятельности и начинает вводить курс валюты заново. Если он подтверждает правильность введенного уже курса, то состояние деятельности переходит к узлу объединения. В этом случае необходимо предельно осторожно подходить к реализации подобной системы подтверждения правильности введенных данных, так как вероятно возникновение ошибки второго рода. Если подтверждение правильности не вводить, то появляется вероятность появления ошибки первого рода.

Под ошибкой первого рода понимается ложная тревога, например, курс валюты введен верно, а установленные критерии защиты не позволяют этот курс установить текущим.

Под ошибкой второго рода понимается пропуск события, например, курс валюты введен неверно, но являясь злоумышленником, оператор, или другое лицо, несанкционированно получившее доступ к АРМ АС банка, подтверждает правильность ввода и устанавливает ложный курс текущим. Необходимость введения действия подтверждения данных при полученной дельта больше допустимой дельты должна исходить из текущей экономической ситуации в стране, о чем было сказано выше.

Итак, остается рассмотреть узел объединения. При правильности введенного курса (достоверность определяется либо автоматически при сравнении полученной дельты ввода с допустимой дельтой, либо непосредственно подтверждением оператором АРМ), определенной несколькими способами, состояние деятельности переходит к узлу объединения, который действительно объединяет доступные пути получения факта достоверности и приводит их к одному результату – переход к

следующему, последнему действию «Установить введенный курс валюты текущим». Это действие происходит без лишних подтверждений, условий, и т.п. и заканчивается конечным узлом (состоянием) деятельности.

Построение субъектно-предикатной модели

Можно исследовать преднамеренную угрозу банковской АС в той же ситуации с вводом курса валюты, но уже на примере субъектно-предикатной модели^[12] (рис. 2).

На первый взгляд модели кажутся отдаленно похожими, сразу бросаются в глаза два конечных узла деятельности, в вербальной модели был один конечный узел. На самом деле модели, и вербальная, и субъектно-предикатная, отображают одну суть.

Прежде чем переходить уже к непосредственному сравнению моделей, необходимо пояснить условные обозначения, используемые на диаграмме. Если в вербальной модели субъектом проводимых операций (протекания процесса деятельности) был оператор АРМ АС банка, который вводил курс валют, сравнивал его со вчерашним курсом, а сам курс выступал объектом деятельности, то здесь в роли субъекта (сущности) выступает курс валюты. Поэтому все действия совершает как бы сам курс (он «содержится в голове», он «является введенным» и т.п.).

Еще одна особенность субъектно-предикатной модели представляет собой охват каждым одним действием всех действий (состояний) деятельности, но в соответствующий каждый отдельный момент рассматривается только действие (в редких случаях действия), выполняемое в данный момент. Остальные действия неактивны в настоящее время, либо они уже совершены, либо не совершены и будут совершены в дальнейшем (такие несовершенные действия обозначаются символом инверсии).

Касаясь самого предмета субъектно-предикатного динамического знания, необходимо условиться, что невозможно отрицание всех действий, ибо если все отрицается, то ничего и нет. Поэтому первым действием после начального состояния диаграммы деятельности является действие курса валют –

«Содержится в голове», которое на первый взгляд может показаться довольно странным. Это действие принимается как неоспоримое начальное действие деятельности.

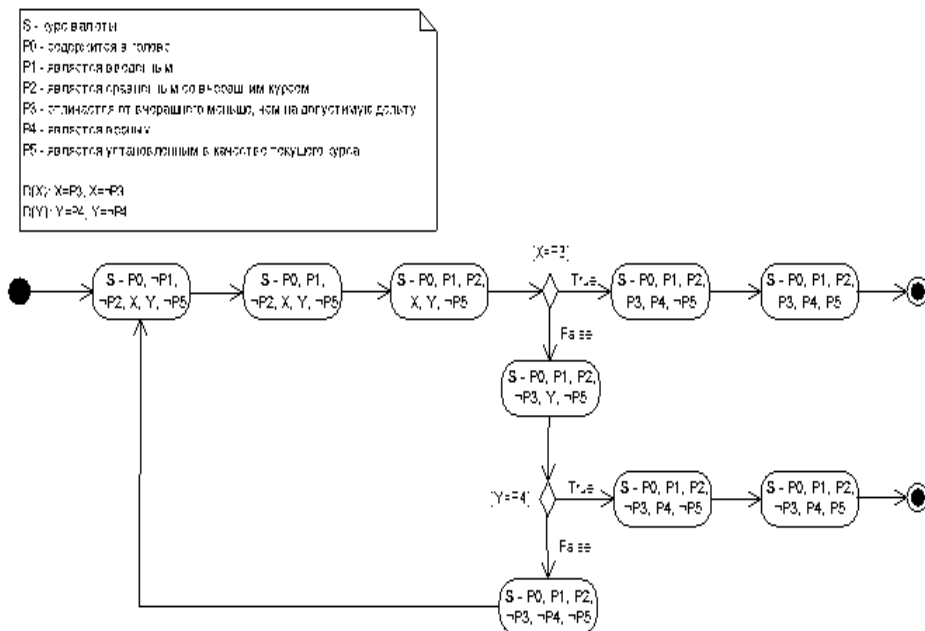


Рис. 2 – Метод диаграмм деятельности.

Субъектно-предикатная модель Символ S является воплощением субъекта (сущности) курса валюты, в символе P находят свое отображение действия. X и Y – действия (состояния), которые могут быть как совершены, так и не совершены в данный момент времени, то есть по сути являются неизвестными состояниями. Это можно объяснить, как если бы чтобы узнать совершено ли это действие или нет, нужно выполнить еще несколько действий. Так действие X включает в себя два состояния: курс валюты «отличается от вчерашнего курса меньше, чем на допустимую дельту» и соответственно, отрицание предыдущего состояния, то есть курс валюты «не отличается...». Действие Y также включает два состояния: курс валюты «является верным» и «не является верным». Естественно,

нельзя заранее определить, является ли курс введенным верно, не проделав предварительно еще несколько действий.

Сравнение вербальной и субъектно-предикатной моделей. Теперь, когда особенности субъектно-предикатной модели рассмотрены, можно приступить к пошаговому сравнению модели с вербальной моделью.

Начальным действием вербальной модели является действие «Ввод курса валюты», в то время, как уже было замечено выше, диаграмма субъектно-предикатной модели начинается с действия курса валюты – «Содержится в голове». Причина такого различия заключается в особой предустановке субъектно-предикатной модели: невозможно отрицать все состояния сущности, иначе тогда ее самой не может существовать.

Далее, после действия ввода курса валюты в вербальной модели следует узел решения, подтверждающий факт ввода данных в АС банка. В случае с языком субъектно-предикатного знания, здесь такой необходимости нет, т. к. в отличие от вербального языка субъектно-предикатный язык является довольно строгим, не допускающим различного рода неоговоренности.

Объекты, «Введенный курс валюты» и «Таблица сравнения» вместе с приложенными комментариями, являются необязательными в вербальной модели, но опять же, используются для наиболее полного представления картины протекающей деятельности.

Действия сравнения и следующие за ними узлы решения, которые в зависимости от результата сравнения определяют дальнейший путь протекания деятельности, идентичны в обеих моделях.

Основное визуальное различие между диаграммами обеих моделей заключается в дальнейших развязках деятельности. В вербальной модели в случае положительного результата сравнения полученной дельты ввода и допустимой дельты, а также в случае подтверждения правильности отрицательного результата сравнения достаточно интегрировать эти потоки управления, исходящие с двух различных узлов, в одно действие,

с помощью узла объединения. Смысл понятен: если курс валюты является верным, то необходимо установить его в качестве текущего курса, что, собственно, и отображает диаграмма. С субъектно-предикатной моделью не все так просто. При проведении сравнения двух значений дельты, полученной и допустимой, на выходе получаются два различных состояния (полученная дельта меньше допустимой, или наоборот), которые в субъектно-предикатном языке уже нельзя объединить в одно состояние. Эта разница возникает в силу того факта, что каждое состояние субъектно-предикатной модели включает в себя все предыдущие и будущие состояния, которые на данный момент являются неактивными, но, тем не менее, они существуют (см. выше). Стоит заметить, что смысл от этого различия в моделях не меняется: и в субъектно-предикатной, и в вербальной модели в случае верно введенного курса он устанавливается в качестве текущего. Перед установкой курса в качестве текущего в субъектно-предикатной модели необходимо дополнительное состояние (действие): курс валюты «Является верным». Данное действие необходимо для того чтобы в случае неверно введенного курса (этот факт выявляется при подтверждении оператором правильности), можно было перейти на действие курса валюты – «Не является верным», уже от которого поток управления перейдет к начальному действию курса валюты – «Содержится в голове» – для введения курса валюты заново. Вербальная модель дополнительных разъяснений в данном случае не требует.

Заканчиваются диаграммы обеих моделей конечным состоянием. Разница лишь в том, что в вербальной модели одно конечное состояние, а в субъектно-предикатной, в силу особенностей ее языка, их два (см. предыдущий абзац).

Несмотря на визуальное отличие диаграмм обеих моделей, практический смысл их представления одинаков, различия же являются отражением особенности их языка (вербальный язык и язык субъектно-предикатного динамического знания).

Заключение. Для подведения итогов проделанной работы необходимо разъяснить практический интерес разработанной модели, а также перспективы их применения.

Ссылаясь на конкретную, разработанную в настоящей работе модель (две получившиеся модели, по сути, являются одной и той же моделью, только представлены с помощью разных «языков» – вербального и субъектно-предикатного), можно сделать вывод, что эта наглядная модель, раскладывающая весь процесс ввода курса валюты в автоматизированную систему банка на минимально возможные единицы деятельности – простейшие действия, облегчает выявление потенциальных угроз на всех этапах этого процесса. Таким образом, ответственное должностное лицо за обеспечение безопасности АС может проследить, какие конкретные действия совершаются оператором при течении процесса ввода курса валюты, и заблаговременно выявить возможные каналы утечки информации. Здесь необходимо смотреть на эту ситуацию с разных сторон, наблюдать и моделировать, что может происходить при различных вариантах развития событий, тем самым предугадывая возможные действия злоумышленника. Соответственно, после этого разработать превентивные меры.

Также разработанную модель можно применять и в других случаях. Следующие примеры показывают актуальность применения этой модели в совершенно разных сферах деятельности:

На производстве для автоматического контроля качества продукции сложных изделий, когда параметр вновь производимого изделия сравнивается с неким допуском, но в некоторых случаях может и выходить за его пределы (но если допуск задан неадекватно, система не распознает опасности брака);

В диспетчерских службах в авиации, когда в некоторых случаях, вероятно, диспетчер может разрешить тому или иному борту манёвр, формально выходящий за рамки допустимого (здесь встает вопрос, насколько корректно задано это самое "допустимое" значение), или в работе автопилота на борту, когда система не "призовет на помощь" пилота, решив, что ситуация не выходит за рамки штатной, в то время как в действительности выходит (в этом случае неверно заданы "рамки");

При анализе большого количества данных (в самых разных областях), когда необходимо выявить и проанализировать "отскоки", нетипичные значения, лежащие вне поля "обычных" значений – границы этого поля задаются принимаемыми допущениями относительно значений статистических параметров (предельно допустимой "дельтой").

Таким образом, можно формально сформулировать тип информационных угроз, описываемых данной моделью. Далее, внутри этого типа можно анализировать опыт из разных областей – и передавать с требуемым обобщением, например, опыт банковских служащих авиадиспетчерам и наоборот.

Далее, если продолжить строить подобные модели одним и тем же методом, можно в результате прийти к формальной классификации информационных угроз в зависимости от свойств модели, описывающей каждый тип – наряду с теми сугубо утилитарными и не всегда безупречными классификациями, которыми сейчас описываются угрозы. А это выводит информационную безопасность на качественно новый уровень.

Список литературы

1. Руководящий документ Гостехкомиссии России «Защита от несанкционированного доступа к информации. Термины и определения». – М.: ГТК РФ, 1992. – 13 с.;
2. Информационная безопасность предприятия: сущность [электронный ресурс]. – режим доступа: <http://1fin.ru/?id=809>, свободный (20.03.15);
3. Доктрина информационной безопасности Российской Федерации. Утверждена Президентом Российской Федерации от 9 сентября 2000 г. № Пр-1897;
4. Википедия. Диаграмма деятельности [электронный ресурс]. – режим доступа: <https://ru.wikipedia.org/wiki>, свободный (20.03.15);
5. Г. Буч, Д. Рамбо, А. Джекобсон. Язык UML - Руководство пользователя. ЧАСТЬ IV - Основы моделирования поведения. Глава 19. Диаграммы деятельности [электронный ресурс]. – режим доступа:

<http://dit.isuct.ru/ivt/books/CASE/case11/ch19.htm> , свободный (20.03.15);

6. Информационная безопасность и защита информации. Учебное пособие. – Ростов-на-Дону, 2004 [электронный ресурс]. – режим доступа: <http://window.edu.ru/library/pdf2txt/482/57482/27741>, свободный (20.03.15).

7. Центр дистанционного образования. Лекция 9. Диаграмма деятельности [электронный ресурс]. – режим доступа: http://edu.dvgups.ru/METDOC/GDTRAN/YAT/ITIS/PROEK_INF_SIS/METHOD/KURS_L/frame/frame_tema9.htm , свободный (20.03.15);

8. Национальный открытый университет Интуит. Лекция 5: Диаграмма активностей: крупным планом [электронный ресурс]. – режим доступа: <http://www.intuit.ru/studies/courses/1007/229/lecture/5958> , свободный (20.03.15).

9. Центральный Банк Российской Федерации. Динамика официального курса заданной валюты [электронный ресурс]. – режим доступа: http://www.cbr.ru/currency_base/dynamics.aspx?VAL_NM_RQ=R01235&date_req1=20.11.2014&date_req2=20.03.2015&rt=1&mode=1 (20.03.15);

10. Википедия. Финансовый кризис в России (2014—2015) [электронный ресурс]. – режим доступа: <https://ru.wikipedia.org/wiki>, свободный (20.03.15);

11. Центр обмена СКВ Пушкинский. Покупка доллара - как формируется курс? [электронный ресурс]. – режим доступа: <http://www.push52.ru/reader.aspx?ModeName=view&TypeNoteName=News&NoteName=pokupkadollara> (20.03.15).

12. Pshenichny, C.A., and Mouromtsev, D.I., 2015. Grammar of Dynamic Knowledge for Collaborative Knowledge Engineering and Representation. In: Diviacco, P., Fox, P., Pshenichny, C., Leadbetter, A. (Eds): Collaborative Knowledge in Scientific Research Networks, IGI Global, pp 323-354.



**ЦИФРОВАЯ ОБРАБОТКА
СИГНАЛОВ И
ИЗОБРАЖЕНИЙ**
СЕКЦИЯ 5

А.Г. Алёхина, В.А. Краснокутский, С.В. Кривошеев
 Донецкий национальный технический университет
 Кафедра компьютерной инженерии
 E-mail: nastyal1994santa@gmail.com

ФИЛЬТРАЦИЯ ШУМОВ ВО ВСТРАИВАЕМЫХ СИСТЕМАХ

Аннотация.

Алёхина А.Г., Краснокутский В.А., Кривошеев С.В.. Фильтрация шумов во встраиваемых системах. В статье рассматриваются вопросы фильтрации шумов во встраиваемых системах для транспортных средств.

Стремительное развитие полупроводниковых приборов, а в дальнейшем и интегральных микросхем привело к появлению нового типа цифровых систем на базе микропроцессорной (микроконтроллерной) техники - встраиваемой системы (англ. EmbeddedSystem).

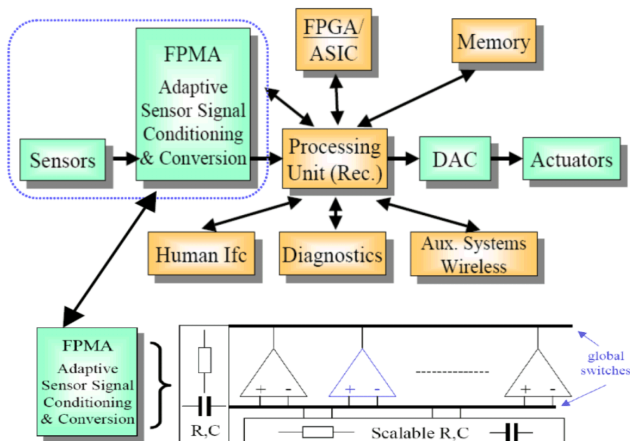


Рисунок 1 – Архитектура встраиваемых систем

Встраиваемая система обладает определенной комбинацией аппаратного и программного обеспечения (рис. 1), предназначенной для выполнения специальных функций и

включает в себя центральный процессор (CPU), датчик (Sensors), ЦАП (DAC), АЦП (ADC) и другие компоненты.

Концепция разработки такой системы заключается в том, что она работает, будучи встроенной в устройство, которым она управляет.

В настоящее время встраиваемые системы получили широкое применение и стали актуальны во многих сферах человеческой деятельности: в промышленном производстве, транспорте, медицинском оборудовании (рис.2). Такое распространение обусловлено широкими возможностями и универсальностью микропроцессорных устройств.



Рисунок 2 – Сферы применения встраиваемых систем
К свойствам встраиваемых систем относят:

- низкое энергопотребление;
- надежность;
- адаптивное поведение;
- самоконфигурирование;
- самовосстановление.

Наиболее широко встраиваемые системы применяются в транспортных средствах (самолеты, суда, автомобили, поезда и т.д.). Например, встроенные микроконтроллеры делают агрегаты автомобиля более безопасными, экономичными, обеспечивающими легкость управления и комфортабельность движения (рис.3), что оправдывает их применение ведущими

автомобильными производителями (BMW, Mercedes-Benz, Infiniti, Acura, Lexus и др.)

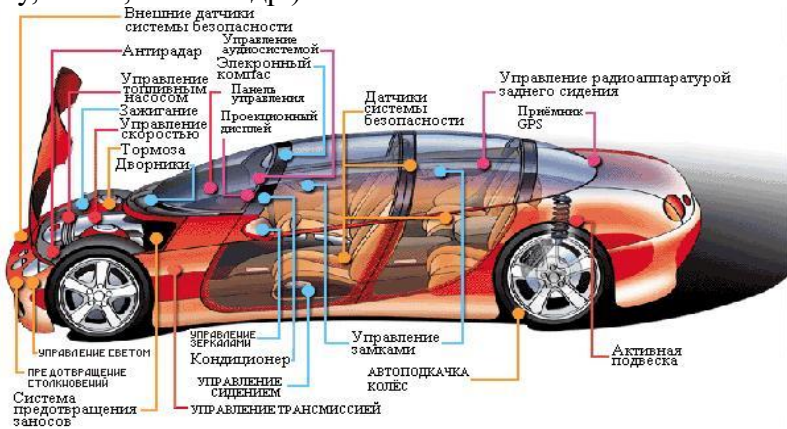


Рисунок 3 – Применение встраиваемых систем в транспортных средствах

При использовании встраиваемых систем для управления транспортными средствами возникает необходимость в передаче информации между датчиками и бортовым компьютером в сложных условиях, которые вызваны большим количеством источников шумов разной природы. Потенциальные источники шума включают в себя импульсные источники питания, цифровой шум от других частей системы и внешние источники. Кроме этого, встраиваемая система должна учитывать специфику объекта управления, чтобы необходимый цикл вычислений укладывался в отведенный временной интервал. Для увеличения уровня безопасности необходимо выполнять разработку с учетом целого ряда критериев: соответствующей вычислительной производительностью микроконтроллера, эффективным по быстрдействию алгоритмом управления, схемами интерфейсов с минимально возможными задержками в передаче сигналов.

Для снижения уровня шума в транспортных средствах используют цифровые фильтры, которые могут изолировать нас от ненужного шума или улучшить необходимый сигнал.

Цифровые фильтры, как правило, бывают с конечной импульсной характеристикой (КИХ) или бесконечной

импульсной характеристикой (БИХ). БИХ-фильтры используют обратную связь для рециркуляции сигнала. КИХ-фильтры являются более стабильными, но имеют более высокую стоимость вычисления и более низкое частотное "разрешение", чем БИХ-фильтров. БИХ-фильтры страдают от фазовых искажений, когда сигналы на разных частотах пересекают фильтр на разных скоростях (аналогично дисперсии волн). КИХ-фильтры, как правило, могут быть разработаны без фазовых искажений.

Цифровые фильтры устраняют различные виды шумовых помех, которые могут возникать из-за временных изменений частотной характеристики, из-за импульсного или теплового шума, источником которого являются переключательные системы. Типы помех разделяют по источникам их возникновения, по энергетическому спектру, по характеру воздействия на сигнал, по вероятностным характеристикам и другим признакам. Поскольку природа этих шумов различна для фильтрации необходимо использовать комбинированную систему фильтров, которая может существенно улучшить общие параметры обработки сигналов.

Таким образом, фильтрация шумов во встраиваемых системах в различных отраслях человеческой деятельности является актуальной проблемой на сегодняшний день и требуется разработка эффективных по различным критериям алгоритмов шумоподавления с целью повышения их качества.

Список литературы

1. П. Эмилиани, В. Каппелини, А. Дж. Константинодис. Цифровые фильтры и их применение/Изд-во Энерготомиздат. 1983 г.

2. <http://www.cs.rice.edu/~keith/EMBED/>:12.05.2015г.

3. P.H. Feiler, B. Lewis, and S. Vestal. The SAE Architecture Analysis and Design Language (AADL) Standard: A basis for model-based architecture-driven embedded systems engineering. In Proceedings of the RTAS Workshop on Model-driven Embedded Systems, pages 1–10, 2003

004.932.72'1

М. А. Варламова

Донецкий национальный технический университет, г. Донецк
факультет компьютерных наук и технологий, кафедра
компьютерных систем мониторинга
E-mail: mavarlamova@mail.ru

ПРИМЕНЕНИЕ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЙ В СИСТЕМАХ ЛОКАЛИЗАЦИИ ЧЕРТ ЛИЦА ЧЕЛОВЕКА

Аннотация.

Варламова М. А. Применение вейвлет-преобразований в системах локализации черт лица человека. Для комфорта и безопасности в различных сферах современной жизни интенсивно используются биометрические технологии. Задача идентификации личности человека относится к разряду плохо формализуемых.

Постановка проблемы. В современном мире биометрия все более активно проникает в различные сферы жизни человека. Значительный интерес к тематике цифровой идентификации личности принято связывать с ростом угрозы международного терроризма, проблемами процессов демократизации при переходе к информационному обществу, а также в связи с увеличением объемов торговых операций, осуществляемых через глобальные компьютерные сети, в частности через Интернет. Проблема создания систем искусственного интеллекта является перспективной на данном этапе развития технологий. Отмечается особый интерес к системам автоматического анализа, позволяющим идентифицировать личность человека по изображению его лица. Данной проблеме уделяется повышенное внимание в связи с необходимостью создания систем мгновенного определения личности человека и беспарольных систем контроля доступа.

Основная часть. Статический метод, основанный на форме лица как объекте идентификации, заключается в создании двух или

трехмерного образа лица человека [3]. Камерой и специализированным программным обеспечением на изображении лица подчеркиваются контуры глаз, губ, бровей, носа и т. д. Затем вычисляют расстояния между этими элементами и прочие параметры. По этим сведениям создается образ, который для сравнения преобразуется в цифровую форму. Этот способ относится к наиболее динамично развивающимся направлениям в индустрии биометрии. Его привлекательность основана на том, что не требуется специального дорогого оборудования. Достаточно персонального компьютера и видеокамеры [1]. Кроме того, отсутствует физический контакт с устройствами. Не нужно прикасаться ни к чему, либо останавливаться, специально ожидая срабатывания системы.

Обычно, процесс распознавания лиц состоит из таких двух этапов, как поиск на изображении области лица и сравнение найденного лица с лицами, которые содержатся в базе данных. С недавнего времени стали широко использовать методы обработки данных, основанные на вейвлет-преобразованиях. Оно показало свою эффективность при решении широкого класса задач, которые связаны с цифровой обработкой изображения.

Вейвлет-преобразование — это [интегральное преобразование](#), которое представляет собой [свертку вейвлет-функции](#) с сигналом. Способ преобразования функции (или сигнала) в форму, которая или делает некоторые величины исходного сигнала более поддающимися изучению, или позволяет сжать исходный набор данных. Вейвлетное преобразование сигналов является обобщением спектрального анализа. В дискретном вейвлет-преобразовании наиболее значимая информация в сигнале содержится при высоких амплитудах, а менее полезная — при низких. Среди достоинств вейвлет-преобразований выделяют такие [4]:

- вейвлетные преобразования обладают всеми достоинствами преобразований Фурье;

- вейвлетные базисы могут быть хорошо локализованными как по частоте, так и по времени. При выделении в сигналах хорошо локализованных разномасштабных процессов можно рассматривать только те масштабные уровни разложения, которые представляют интерес;

- базисные вейвлеты могут реализоваться функциями различной гладкости.

Однако можно выделить один недостаток, а именно – относительная сложность преобразования.

Вейвлет-преобразование используют для получения характеристик изображения. Коэффициенты вейвлет-преобразования несут в себе информацию об анализируемом процессе и о том, какой вейвлет используется. В зависимости от того, какую информацию необходимо извлечь из процесса, делается выбор анализирующего вейвлета. Иногда, используя разные вейвлеты, можно полнее выявить и подчеркнуть те или иные свойства анализируемого процесса. Это возможно благодаря тому, что каждый вейвлет имеет характерные особенности во временной и частотной областях.

Выделяют группу классических функций, которые используются в вейвлет-анализе. Это такие функции, как вейвлет Хаара, вейвлет Добеши, вейвлет мексиканская шляпа, вейвлет Морле и прочие. Примитивы Хаара относятся к таким функциям для двумерного пространства.

Для обработки растровых изображений, которые являются дискретным сигналом, необходимо применять двумерные дискретные вейвлет-преобразования. Многомасштабный анализ (multiresolution analysis, MA) должен строиться в пространстве функций двух переменных. Самый распространенный способ – это применения тензорного произведения одномерных MA. В качестве двумерной скейлинг-функции берется (1):

$$\Phi(t, s) = \varphi(t)\varphi(s). \quad (1)$$

Вместо одного вейвлета возникает три (2):

$$\Psi_{LH}(t,s)=\varphi(t)\varphi(s), \Psi_{HL}(t, s) = \varphi(t)\varphi(s), \Psi_{HH}(t, s) = \varphi(t)\varphi(s), \quad (2)$$

где L означает низкую частоту, H – высокую.

За один шаг двумерное вейвлет-преобразование выделяет одну низкочастотную и три высокочастотных компоненты исходного сигнала-изображения. На рисунке 1 представлен результат [2] одного шага прямого вейвлет-преобразования, а именно – преобразования Хаара.

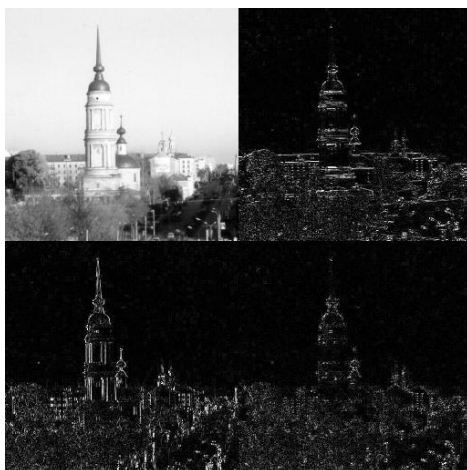


Рисунок 1 – Один шаг преобразования Хаара

В противовес, на рисунке 2 представлен результат выполнения преобразования Фурье. Для данного примера был использован FFT-алгоритм быстрого преобразования Фурье [5]. На данном примере видно, что применение преобразования Фурье для выделения ключевых характеристик лица человека значительно уступает в информативности преобразованию Хаара и вейвлет-преобразованиям в целом.



Рисунок 2 – Результат преобразования Фурье

Высокочастотные составляющие можно использовать для выделения перепадов, контуров изображения, участков резкой смены цвета или яркости. На рисунке 3 после обработки с помощью вейвлетов (Добеши D4) более ярким цветом

обозначены коэффициенты большей амплитуды. Четко видно, что их положение указывает на резкие перепады яркости. Такие перепады являются наиболее информативными при беглом просмотре любого изображения. Вейвлет-представление позволяет их локализовать путем последовательного уточнения, начиная с более крупных масштабов.

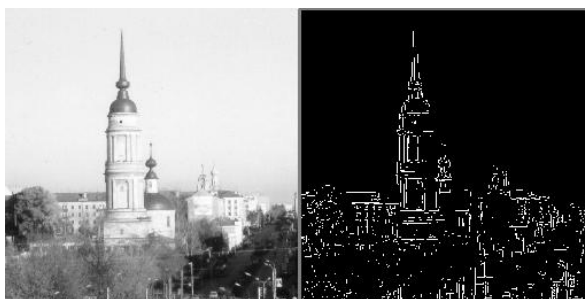


Рисунок 3 – Обработка с помощью вейвлета Добеши D4

Выбор конкретного вида и типа вейвлетов во многом зависит от анализируемых сигналов и задач анализа. При практическом использовании вейвлетов необходимо уделять достаточное внимание проверке их работоспособности и эффективности для поставленных целей по сравнению с известными методами обработки и анализа.

Несмотря на то, что доминирующим способом идентификации личности остается распознавание отпечатков пальцев, идентификация личности по изображению лица все более стремительно набирает обороты. Основные прогнозы сводятся к тому, что внедрение биометрических устройств безопасности в скором будущем приобретет лавинный характер. Борьба с глобальным терроризмом потребует практического использования любых достижений в этой сфере. Благодаря интенсивному развитию мультимедийных и цифровых технологий и дальнейшее их удешевление позволят разработать и внедрить данные системы идентификации. Возможно, в скором времени они перейдут от научных исследований к коммерческим технологиям.

Эффективно распознавать лицо можно только в определенных условиях, именно поэтому крайне важно при внедрении биометрии лица понимать, в каких условиях будет эксплуатироваться система. Для повышения эффективности распознавания лица в идентификационных зонах следует организовывать направленный поток людей (дверные проемы, рамки металлодетекторов, турникеты и т.п.) для обеспечения возможности кратковременной (не более 1–2 с) фиксации лица каждого посетителя.

Выводы. Рассмотренные вейвлеты являются лишь малой частью от того множества вейвлетов, которые были предложены исследователями. Это означает, что работы в данном направлении ведутся по сей день. Потому как не создана оптимальная модель, которая позволяет идентифицировать личность по изображению лица, а лишь накоплен огромный опыт эвристического решения отдельно стоящих задач. Необходимо стремиться к созданию надежной и уверенной модели, потому как предпосылки для создания такой модели в будущем можно наблюдать уже сегодня.

Список литературы

1. Самообучающиеся системы [электронный ресурс] // http://gendocs.ru/v32756/самообучающиеся_системы (дата последнего обращения: 15.04.2015).
2. Л. Левкович-Маслюк, А. Переберин. Вейвлет-анализ и его приложения [электронный ресурс] // http://algolist.manual.ru/compress/image/leo_lev/lecture3/wav3_0.php (дата последнего обращения: 29.04.2015).
3. Способы идентификации личности человека [электронный ресурс] // <http://mirprognozov.ru/prognosis/107/804/> (дата последнего обращения: 29.04.2015).
4. Вейвлет-преобразование [электронный ресурс] // <http://ru.wikipedia.org/wiki/Вейвлет-преобразование> (дата последнего обращения: 25.04.2015).
5. Павлейно М. А. Спектральное преобразование в MATLAB: учеб.-метод. пособие, СПбГУ, 2007 – 150 стр.

УДК 519.725, 681.3

В.О. Дяченко, О.Н. Дяченко

Донецкий национальный технический университет

ОСОБЕННОСТИ ПРИМЕНЕНИЯ ДВОЙСТВЕННЫХ ПОЛИНОМОВ ДЛЯ АППАРАТНОЙ РЕАЛИЗАЦИИ ЦИКЛИЧЕСКИХ КОДОВ

Аннотация

Дяченко В.О., Дяченко О.Н. Особенности применения двойственных полиномов для аппаратной реализации циклических кодов. Предлагается применение двойственных полиномов для аппаратной реализации циклических кодов. Рассматриваются особенности альтернативного построения кодирующих и декодирующих схем.

Внедрение инновационных технологий во всех областях человеческой деятельности приводят к непрерывному увеличению объема накапливаемой информации. Все большее значение приобретают способы помехоустойчивого кодирования, обеспечивающие требуемую достоверность при передаче, обработке и хранении информационных данных. Одними из наиболее эффективных для исправления ошибок и, в особенности, пакетов ошибок, являются циклические коды. В настоящее время существует широчайший спектр разработанных и уже успешно используемых на практике кодов [1-5]: циклические коды Хэмминга, БЧХ, Файра, Рида-Соломона и многие другие. Наиболее известные примеры: (255, 223, 33) код Рида-Соломона для космической связи NASA, укороченные коды Рида-Соломона над полем Галуа GF(28) для CD-ROM, DVD и цифрового телевидения высокого разрешения (формат HDTV), расширенный (128, 122, 7) код Рида-Соломона над полем Галуа GF(27) для кабельных модемов, (255, 239) код рекомендован в качестве внешнего кода в WiMax. Кроме того, циклические коды можно использовать не только для помехоустойчивого

кодирования при передаче данных, а также везде, где есть необходимость в предотвращении искажения информации, например [2]: обнаружение и исправление ошибок в поврежденных или дефектных носителях информации; обнаружение и исправление ошибок при умышленном изменении информационных сообщений с целью дезинформации; обнаружение и исправление модификации информации об авторе или исполняемого кода с целью «взлома» программного обеспечения; защита программного обеспечения или данных от копирования с лицензионного диска; восстановление одного или нескольких томов многотомного архива, искаженных или вообще потерянных при загрузке из сети; обнаружение и исправление ошибок в цепочках ДНК в геномной инженерии. Поэтому вопросы построения и аппаратной реализации циклических кодов являются актуальными, учитывая все большую их востребованность для различных сфер применения.

Естественно, что наиболее популярны в настоящее время коды, исправляющие пакеты ошибок. Тем не менее, циклический код Хэмминга, исправляющий одиночные ошибки, заслуживает особого внимания, поскольку является фундаментом для понимания принципов построения более мощных кодов. Одним из примеров может служить аппаратная реализация декодеров двоичного кода Хэмминга и недвоичного кода Рида-Соломона, исправляющего одиночные (недвоичные) ошибки. Кроме того, аппаратная реализация методов компактного тестирования, в частности, генераторы псевдослучайных тестовых последовательностей и сигнатурные анализаторы, построенные на основе декодеров Хэмминга, используются во многих схемах со встроенным самотестированием, в частности, в микропроцессорных СБИС ведущих зарубежных фирм: Pentium Pro (Intel Corporation); S/390 (IBM); Power PC; MC 202-206 (Motorola); AMD-K6 (Advanced Micro Devices).

При построении циклических кодов во многих случаях приходится их укорачивать [1-3]. В данной работе предлагается для кодирования и декодирования кодов применение двойственных полиномов, что в дальнейшем может дать

преимущества при реализации укороченных кодов.

Из любого (n, k) циклического кода можно получить $(n-i, k-i)$ укороченный код, где n - длина кода, k - количество информационных символов, $i < k$ - параметр укорачивания. Одним из способов декодирования укороченных кодов является использование декодеров, построенных для кодов максимальной длины. При этом принятому кодовому слову предпосылаются i нулей, которые кодером не передаются в канал связи. Недостатком такого способа декодирования является несогласованность скоростей передачи кодером кодового слова (длина такого слова $n-i$, поскольку нули не передаются) и обработки декодером принятого дополненного нулями кодового слова длины n . Кроме того, для формирования синдрома в этом случае необходимо n тактов работы декодера, в то время как при применении другого способа декодирования для этого достаточно $n-i$ тактов.

В отличие от декодера кода максимальной длины, который для формирования синдрома выполняет операции умножения принятого слова на полином X^p и деления на порождающий полином, декодер укороченного кода умножает на полином, равный остатку от деления полинома X^{p+i} на порождающий полином, и полученное произведение делит на порождающий полином. Однако в случае очень большого параметра укорачивания довольно сложно получать остаток от деления полинома X^{p+i} на порождающий полином.

Основная идея отличия применения двойственных полиномов для кодирования и декодирования циклических кодов заключается в том, что декодер выполняет исправление принятого слова по принципу LIFO, а не FIFO, то есть, в обратном порядке следования кодового слова.

Полином $K^*(X) = X^{\deg K(X)} * K(X^{-1})$ называется двойственным полиномом по отношению к полиному $K(X)$.

Коды двойственных полиномов имеют одинаковые характеристики, в частности, одинаковые корректирующие способности, избыточность, аппаратные затраты схемной реализации кодеров и декодеров, быстродействие. В случае

разных полиномов получаем декодер, который обрабатывает биты кодового слова по принципу “последний пришел – первый вышел”.

Пример. Построим два поля Галуа GF(8) как расширения поля GF(2) над примитивными полиномами $p(z)=z^3+z+1$ и $p^*(z)=z^3+z^2+1$ (табл. 1). Элементы поля могут быть представлены в различном обозначении и для ненулевых элементов со степенью большей степени порождающего полинома следуют в обратном порядке.

Таблица 1

Поля Галуа GF(8) с двойственными порождающими полиномами

$p(z)=z^3+z+1$			$p^*(z)=z^3+z^2+1$		
В виде степени	В виде полинома	В двоичном виде	В виде степени	В виде полинома	В двоичном виде
0	0	000	0	0	000
α^0	1	001	α^0	1	001
α^1	z	010	α^1	z	010
α^2	z^2	100	α^2	z^2	100
α^3	$z+1$	011	α^3	z^2+1	101
α^4	z^2+z	110	α^4	z^2+z+1	111
α^5	z^2+z+1	111	α^5	z^2+z	110
α^6	z^2+1	101	α^6	$z+1$	011

Рассмотрим более подробно работу кодера и декодера для кода Хэмминга (7,4) с порождающим полиномом кодера $K(X) = X^3 + X + 1$ и порождающим полиномом кодера $K^*(X) = X^3 + X^2 + 1$.

Пусть информационные символы $A = 1101$ в двоичном виде, или $A(X) = X^3 + X^2 + 1$ в полиномиальном виде. Для систематического кода кодер циклического кода Хэмминга выполняет операцию умножения информационной последовательности на полином X^{n-k} и деление на порождающий полином. Полученный остаток от деления представляет собой

проверочную часть кодового слова. После выполнения этих операций получаем систематический код $X^6 + X^5 + X^3 + 1 = 1101001$, где первая часть 1101 информационная, а вторая 001 – проверочная.

Рассмотрим различные варианты применения порождающих полиномов. В таблице 2: A – кодовое слово, $\Phi C1(A)$ – формирователь синдрома для $K(X)$, A^* – кодовое слово в обратном порядке следования, $\Phi C2(A^*)$ – формирователь синдрома для $K^*(X)$, A^*_7 – кодовое слово в обратном порядке следования с ошибкой в 7-м символе кодового слова A , E^*_7 – полином ошибки, $\Phi C2(E^*_7)$ – формирователь синдрома для $K^*(X)$.

Таблица 2

Различные варианты применения порождающих полиномов

№	A	$\Phi C1(A)$	A^*	$\Phi C2(A^*)$	A^*_7	$\Phi C2(A^*_7)$	E^*_7	$\Phi C2(E^*_7)$
	1	2	3	4	5	6	7	8
1	1	100	1	100	0	000	1	100
2	1	110	0	010	0	000	0	010
3	0	011	0	001	0	000	0	001
4	1	011	1	001	1	100	0	101
5	0	111	0	101	0	010	0	111
6	0	101	1	011	1	101	0	110
7	1	000	1	000	1	011	0	011
					$*X$	100	$*X$	100
					$*X^2$	010	$*X^2$	010
					$*X^3$	001	$*X^3$	001

ΦC представляют собой регистры сдвига с линейной обратной связью (РСЛОС), выполняющие функцию деления кодового слова на порождающий полином. В 5-8 столбцах в дополнительных строках выполняется умножение последовательностей и значение ΦC на X , X^2 , X^3 , а в общем случае – на X^{n-p} (ΦC для декодера Меггитта). Анализ таблицы 2 показывает, что, во-первых, результат деления A на $K(X)$ и A^* на $K^*(X)$ равны нулю (остаток от деления кодового слова, записанного в разном порядке и деленного на двойственные полиномы равен нулю, что означает отсутствие ошибки). Во-

вторых, результаты деления реальной последовательности A^*_7 с ошибкой и последовательности ошибки $E^*_7 = A + A^*_7$ совпадают, что позволяет рассматривать только последовательности ошибок вне зависимости от реальных значений кодовых слов (такой же результат будет для любых одиночных ошибок). В-третьих, при реализации ФС в виде РСЛОС, выполняющего функцию умножения на полином X^{n-p} и деления на двойственный полином последовательности кодового слова в порядке обратного следования дает остаток из всех нулей и последней единицы (как и в обычном декодере Меггитта), что дает преимущества при построении схемы исправления в декодере.

Таким образом, при предлагаемой схемной реализации циклических кодов кодер остается прежним. Декодер имеет следующие отличия. В своем составе он содержит реверсивный циклический n -разрядный буферный регистр, формирователь синдрома и схему исправления ошибки. За первые n тактов принятое кодовое слово записывается в буферный регистр и одновременно формируется синдром. Затем направление сдвига в буферном регистре изменяется и за следующие n тактов с помощью формирователя синдрома и схемы исправления декодер устраняет ошибку и кодовое слово уже без ошибки перезаписывается в буферный регистр. За третьи n тактов направление сдвига в буферном регистре снова меняется и на выходе появляется исправленное кодовое слово.

1. Richard E. Blahut. Algebraic Codes for Data Transmission/ Cambridge University Press, 2012. – 498 p.
2. Рахман П.А. Основы защиты данных от разрушения. Коды Рида-Соломона/ Интернет-ресурс. – Режим доступа: URL <http://icc.mpei.ru/documents/00000885.pdf> Загл. с экрана.
3. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. – М.: Мир, 1976. – 595 с.: ил.
4. Дяченко В.О., Зинченко Ю.Е., Дяченко О.Н. Исследование способов проектирования кодов Рида-Соломона // Інформаційні управляючі системи та комп'ютерний моніторинг (ІУС КМ-2014) : V Всеукраїнська науково-

технічна конференція студентів, аспірантів та молодих вчених, 22-23 квітня 2014 р., м. Донецьк : зб. доп. / Донец. націонал. техн. ун-т; редкол. В.А.Світлична. – Донецьк: ДонНТУ, 2014. – в 2 тт. – т.2. – С. 72-78.

5. Дяченко В.О., Дяченко О.Н. Анализ способов реализации кодов Рида-Соломона, исправляющих двойные ошибки // Современные тенденции развития и перспективы внедрения инновационных технологий в машиностроении, образовании и экономике: материалы Международной научно-практической конференции (Азов, 19 мая 2014 г.). – Ростов н/Д, ДГТУ, 2014.– С. 18-22.

УДК 004.7+004.8

К.С. Лащенко, Н.Г. Соколов, О.А. Гудаев
Донецкий национальный технический университет

ОНТОТЕКСТОЛОГИЯ ИНТЕРНЕТА

Аннотация.

К.С. Лащенко, Н.Г. Соколов, О.А. Гудаев.
Онтотекстология интернета. Представлена новая структура организации знаний и различного текстового материала в сети Интернет, которая позволит существенно улучшить использование памяти серверов путем разумного выбора текстового материала и удаления повторяющихся текстов среди множества сайтов.

Всем нам привычное «облако тегов» отражает общественное предпочтение по использованию слов в подборке текстов. Облако тегов строится на основе колоссальных затрат на память и вычисление селективной функции, которую выполняет компьютерная система. Как было бы просто отличать популярную книгу в библиотеке по лацкану обложки, натёртого до блеска, по сравнению с рядами книг, притрушенных пылью. Разветвлённая компьютерная сеть семантического Интернета собирает статистику хронологии обращения читателя к тексту. Препарируя эмоции, автоматическая функция конструирования «облако тегов» подменяет назначение ключевых слов текста. Допустим, автор текста явно выпишет ключевые слова. Автор может субъективно заблуждаться, что помог читателю понять текст. Генератор текста, расставляя акценты, далеко не поможет понять суть идеи. Понимание текста зависит от уровня развития мышления читателя или склонности его интересов. Медик с трудом поймёт текст по химии. Научно выражаясь: репрезентация получаемых из текста знаний отражается в «картину мира» читателя. Встраивание знаний осуществляется в модель мышления. Разные модели мышления химика и медика имеют

характер абсолютно разных взаимоотношений. Уместно предположить, что мышление – это глобальный процесс, а «картина мира» - это динамическая память мозга, изменяющийся, на лету, локальный процесс мышления. В итоге можно сказать, что «Облако тегов», автоматически порождённое нам в помощь, навязывает нам извне сюрреалистическую точку зрения, а ключевые слова автора текста недостаточно объективны. Автор текста не объективен: он знает о смысле текста всё, но не знает, как донести эту суть читателю, не имея представления о его модели мышления. На что же тогда опереться в понимании текста? Решение будет очевидно, если глубоко задуматься:

- о форме и структуре текста, как входных данных для анализа;

- о процессе написания текста автором;

- о скрытом, тайном смысле высказанной автором идеи, но не раскрытой до конца в тексте.

Самое трудное в поисках решения данной проблемы – это, донести до читателя скрытый, тайный смысл высказанной идеи, которую автор не смог раскрыть в тексте до конца. Таким образом, получается ситуация «детерминированного хаоса в квадрате»: хаос модели знаний читателя о рассматриваемом вопросе в тексте, перемноженный на хаос изложения автором сути идеи, в виде формального текста. В вопросе «скрытых» смыслов существует две гипотезы.

Первая гипотеза заключается в следующем: первоисточник зарождения мысли пытается построить объёмную картину мира рассматриваемого предмета, оперируя простыми понятиями и устанавливая максимальное количество реферативных связей на момент написания текста. Достоинство подхода, изложенного в гипотезе – это понятность. Недостаток – это устаревание реферативных ссылок, в чём и заключается исторический процесс эволюции идеи, заложенной в тексте.

Вторая гипотеза: эволюция первоисточника протекает не в исторической хронологии, а в логической.

Уточним, данная гипотеза предполагает, что первоисточник может эволюционировать в терминах классической логики, а

может «нарушать симметрию» и эволюционировать по правилам неклассической логики. Поясним «нарушение симметрии»: не диалектический материализм; не развитие по спирали; не выполнение закона отрицания отрицаний.

Такая сложная ситуация с неклассической логикой развития первоисточника идеи может называться «информационной волной». Здесь стоит сразу разобраться и получить ответ на вопрос «Почему процесс не “исторический”?» Ожидается, что высказанная идея неизменна во все века, а исторически только уточняется и раскрывается новыми прецедентами применения на практике. То есть имеет хронологию применения, а суть идеи остается неизменной, пройдя сквозь века. Но это не так!

Тогда возникает еще несколько вопросов. Как же происходит трансформация идеи на самом деле? И кто привносит вклад в эти нарушения? Процесс эволюции «живой»! Чередование формализации писателя и интерпретации читателя приводит к разрыву первоисточника с сутью происходящего в современности, в виду желания устанавливать новые реферативные ссылки. Ссылки имеют субъективную окраску, так как их устанавливают новые исследователи текста.

Требование идти со временем «в ногу» заставляет экспериментаторов пересказывать идею первоисточника в понятных современнику терминах, ссылаясь на существующие прецеденты и теории. Тогда искажения неизбежны. Так возникает «информационная волна». Например: «писк» - это противный скрип лебёдки или модный тренд, вызывающий воодушевление.

Можно предположить, что сама современность привносит искажения в первоисточник. При смене хронологии событий искажение изначального изложения онтологий сильнее раскрывают их суть непосредственно для современных читателей. Текст будет обрастать все новыми и новыми искажениями. Прохождение онтологий через хронологию событий можно наблюдать на рисунке 1. В какой-то определенной хронологической последовательности, современный автор может интерпретировать текст добавляя

новые или отнимая старые факты. Таким образом, имеем искажения полученные на основе прецедентов.



Рисунок 1 – Трансформация онтологий текста

Почему происходит искажение смысла первоисточника мысли в ходе передачи информационной волны? А может это новые знания, не сопоставимые с источником. Так происходит из-за сущности формы изложения текста на естественном языке. Форма изложения текста подчинена главному постулату: аргументы и факты облакаются в эмоциональный контекст, чтобы быть интересным читателю. Нудный и монотонный слог усыпляет сознание и блокирует размышления над прочитанным. Человек на подсознательном уровне начинает блокировать получаемую информацию, считая ее угрозой или бесполезным засорением памяти. Даже если сама суть идеи важна и интересна читателю, он не сохранит и не поймет ее из-за неправильной подачи автором. В ходе передачи информационной волны удачный художественный контекст превалирует над логикой излагаемых фактов. Пересказчик интерпретирует понравившийся текст, по своему мнению, вкладывая в контекст новое звучание, превалирующее над первоначальным замыслом, но это не гарантирует мотивированного понимания идеи всеми читателями. Все люди разные, что интересно одному, другой сочтет за мусор.

Поэтому, перед пересказчиком текста стоит практически нереализуемая задача – донести суть идеи до всех читателей. Так бывает, что перевод с японского в несколько раз искуснее оригинала.

Как же найти предел психологического барьера «скучно – не понимаю» / «нравится – всё понятно!»?

Текст, проходя через субъективные точки зрения, нелинейно трансформируется. Это реалии жизни помогают всему

человечеству «не закиснуть». Может ошибаться пересказчик? Да. Его семантическая ошибка в тексте принадлежит иррациональному миру. Но иррациональный мир – источник новых знаний [1, с. 225]. Как делать полезные ошибки? Ответ: фантазия и воображение. Молодая теория воображения не может дать рекомендации на все приёмы художественного оформления текста [2]. Но кое-что может сказать о форме и структуре текста. В компьютерном мире Интернета существует простая формула: контент – это текст плюс контекст. В качестве скрытых тегов выступают ключевые слова электронных документов, заданные для SEO-оптимизации, и автоматически подобранные поисковой машиной аннотированные ссылки. Теория воображения рассматривает процесс создания текста. Поэтому, косвенно затрагивает литературную структуру упаковки смысла в текст. Например, моделью структуры будет <Пейзаж, Персонаж, Правила жизни, Сцена>. Сцена – это полигон, где персонажи проживают явления или ситуации по своим правилам жизни в заданной локации. Структура без формы «рассыплется». В логике воображения не рассматривается низкоуровневая форма формализации художественного замысла. Может быть – это путь развития молодой науки.

Рассмотрим формы изложения онтологий в виде текста. Эти формы не премудрости копирайта. Онтологии «живут» в некоторой форме изложения текста и программируют смысл прочтения. Для конкретного психологического типа читателя устанавливается баланс между любимой формой изложения, её должно быть побольше, и понятной, которая не должна раздражать, а приятно удивлять, её должно быть не много. Стоит перечислить все формы изложения текста по категориям в таблице 1.

Таблица 1 – Формы изложения текста

Форма	Категория	Конкретизация
Имя, название, безусловный адрес, именование	Формальная	Таксономия, термин
Перечень целей,	Формальная	Мотивация:

целевая мотивация, назначение		{принципы, законы}, инструкция, приказ
Перечисление результатов	Взаимосвязь с другими объектами	Модель черного ящика, через входные и выходные данные, плоды, влияние на внешнюю среду
Область применения	Взаимосвязь с другими объектами	Использование, условия внедрения
Функционирование	Процедурная	Порождающая процедура, схема функционирования, суперпозиция функций
Рецепт, сценарий поведения, алгоритм синтеза	Процедурная	Организация процесса, логистика, практический опыт
Идеальная модель	Структурная	Синтез, идеализация через обобщение
Аналогия	Структурная	Прообраз, искусственное сочетание
Перечисление свойств, характеристик	Структурная	Косвенная адресация, ассоциативный ключ
Множество вопросов	Художественная	Интервьюирование
Отрицание свойств	Художественная	Отсечение известных знаний
Метафорический образ	Художественная	Мифотворчество, мистификация, иносказательный образ

Форма – это мощный инструмент программирования извлечения смысла из текста читателем. Форма принимает вид опосредованной интерактивной игры типа ФЕС, без возможности запроса на повторную передачу смысла текста в другой форме.

Если писатель не играет с читателем, то его текст скучен и монотонен, что в свою очередь включает защитный механизм читателя воспринимать информацию как враждебную. Писатель должен уметь создавать разнообразные формы текста. Компьютерные чат боты должны уметь различать все формы текста, чтобы пройти тест Тьюринга. Например, название текста в форме абсолютной адресации «Lexicode for Augmented Reality» вызовет бурю эмоций у передовых учёных и текст надо будет писать впервые [3].

Чем более схожи идеи, тем они дороже как мнение. Коммерческая модель онтологического Интернета – это когда плата берётся за внесение в хранилище. Анализ текста и поиск схожести будет осуществлять данная модель Интернета, в чем проявляется ее несомненное преимущество над современной моделью. Пример адресной строки приведен на рисунке 2.

Поисковый запрос и представление запроса на естественном языке приведено ниже:
Я. {Олег}. Наука. {Коды. [телекоммуникацион.]} | Публикации. {статьи}. Поиск. {маркеры}. «Научные статьи Олега по теме телекоммуникационные коды. Ключ для поиска - “маркеры”».

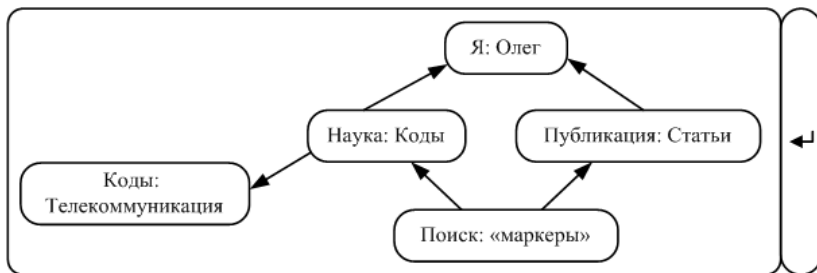


Рисунок 2 – Адресная строка

Наслоение компьютерных фактов теряется в большом количестве сведений. Любая система накопления зависит от желания сохранить всё на потом. Репозиторий данных разрастается в необъятное хранилище. Хранение контекста добавляет широту обзора сведений, но это ведёт к ещё большему

разрастанию данных. Избавится от контекста невозможно. Вне контекста сведения малозначительны. Легкость, с которой формируются слои доказательных материалов, делает затруднительным поиск истины. Контекст для несвязанных фактов страдает многократным повторением одного и того же. Автор не умеет выделить главное и внести это правильно в виде контекста в текст. Мыслить многомерным миром онтологий только для того, чтобы дополнить сообщение контекстом, автору идей кажется невероятной работой с субъективным конечным результатом. Контекст нужен только в будущих периодах использования! Поэтому, автор современного Интернета, не мотивирован и считает работу по созданию текста с онтологическим контекстом бессмысленной. Для примера важности контекста, составим граф развития гипотезы (см. рис. 3). Имя A1 – это автор, первоисточник мысли, который выдвинул гипотезу M1. Авторы: {A1, A2, A3, A4}. Идеи: {M1, M2, M3, M4, M5, M6, M7}.

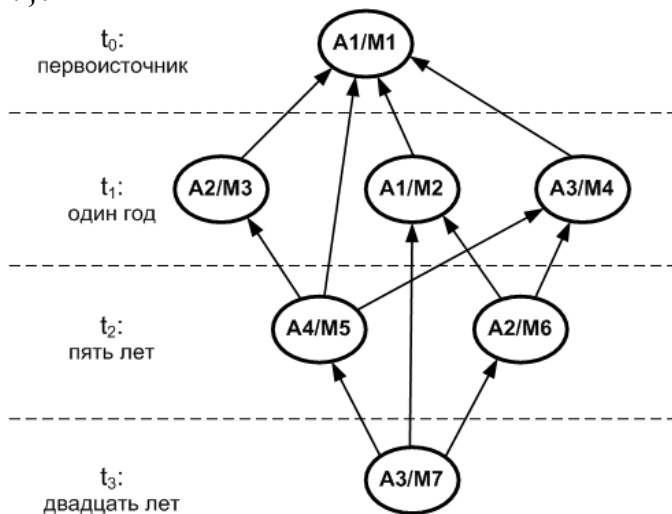


Рисунок 3 – Граф развития гипотезы

Чтобы не засорять хранилище однотипными текстами, плата за текст схожий с уже имеющимися текстами берется намного

выше, чем если бы текст представлял собой большую уникальность по сравнению с остальными в хранилище. Это будет сильнее мотивировать автора к написанию более уникальных текстов и экономить огромное количество пространства.

Выводы. Таким образом, данное представление организации Интернета позволяет создать более связную по смыслу онтологическую систему, которая бы могла учитывать различные формы изложения текста для одной и той же идеи. Такая подача информации позволит донести до любого читателя суть идеи ввиду наличия огромного множества форм изложения текста. В свою очередь у авторов или пересказчиков текста сформируется своя аудитория читателей, а у читателей сформируется ряд любимых авторов, понравившихся по форме изложения идей в тексте. Ввиду вышесказанного, существенно повысится мотивация по созданию доступного изложения любой идеи, мысли, гипотезы. В результате компьютерная система онтологической организации Интернета сможет: переформулировать текст, организовывать обратную связь вовлеченных авторов, вопреки современной ленточной подачи параллельных комментариев, осуществлять создание исторической хронологии всегда связанных первоисточников.

1. Ракитов, А.И. Философия компьютерной революции [Текст] / А.И. Ракитов. – М. : Политиздат, 1991. – 287 с. – ISBN 5–250–01308–2.

2. Шрагина, Л.И. Логика воображения [Текст] : Учебное пособие / Л.И. Шрагина. – 2-е изд., дороб. – М. : Народное образование, 2001. – 192 с.

3. Гудаев, О. Иероглификация вселенского масштаба [Текст] / О. Гудаев // Наука и техника. – 2014. – № 7(98). – С. 7–10.

УДК 004.925

Лунтовская А.А.

Донецкий национальный технический университет
кафедра прикладной математики и информатики
E-mail:anastasia.luntovskaya@gmail.com

ПРОЦЕДУРНАЯ ГЕНЕРАЦИЯ ОБЛАКОВ И РЕНДЕРИНГ НЕБА В РЕАЛЬНОМ ВРЕМЕНИ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКОГО ПРОЦЕССОРА

Аннотация

Лунтовская А.А. Процедурная генерация облаков и рендеринг неба в реальном времени с использованием графического процессора. В статье рассмотрен процесс генерации и рендеринга облаков и реализация цикла смены дня и ночи для визуализации динамического 3D окружения в компьютерных играх с использованием графического процессора. Генерация текстуры облаков выполнена с помощью алгоритма шума Перлина. Алгоритм реализован в Unity3D на языке шейдеров Cg.

Постановка проблемы.

Генерация и представление облаков является одной из основных проблем рендеринга реалистичного неба. Для создания неба в играх традиционно используются два основных подхода. В статическом подходе одна или больше фотографий текстурируются на 3D модель неба. Например, текстура облаков может накладываться на голубую текстуру неба. Несмотря на то, что само небо в результате таких действий может быть представлено очень убедительно и реалистично, учитывая качество используемых фотографий, отсутствие динамики является существенным недостатком такого подхода. Второй подход предполагает процедурную генерацию облаков. Объемные методы пытаются моделировать 3D пространство, занимаемое облаком, плоскостные представляют облака с помощью 2D

текстуры, накладываемой на 3D модель неба. При условии, что зритель всегда остается на уровне земли, такой метод вполне может использоваться для создания реалистичного неба. В статье будет рассмотрен алгоритм генерации 2D текстуры облаков, которая впоследствии накладывается на купол неба.

На сегодняшний день алгоритм шума Перлина широко применяется для процедурной генерации туманов, облаков, огня, различных эффектов поверхности воды и земли. Существует множество реализаций шума Перлина на центральном процессоре, но с увеличением мощности графических процессоров становится актуальной реализация алгоритма на ГПУ, что значительно снижает временные затраты, поскольку алгоритм включает в себя достаточно большое количество вычислений, которые могут выполняться параллельно.

Анализ исследований и публикаций.

Кен Перлин [1] в 1983 году предложил алгоритм генерации процедурной текстуры псевдослучайным способом, который в компьютерной графике используется для увеличения реализма поверхности геометрических моделей.

Функция шума имеет несколько важных характеристик:

- для каждой входной позиции вычисляется повторяющееся псевдослучайное значение;
- диапазон генерируемых значений известен и чаще всего принимается за $[-1,1]$;
- очевидные повторяющиеся узоры исключаются;
- пространственная частота инвариантна смещению [2].

Реализация шума Перлина в пиксельном шейдере имеет ряд преимуществ:

- требуется меньше памяти для текстур;
- текстура имеет высокую периодичность, шаблон узора не повторяется часто;
- значительно превосходит результаты ЦПУ по времени;
- более высокое качество интерполяции, чем то, что доступно при аппаратной фильтрации текстур, таким образом, результат

получается более сглаженным;

- появляется возможным создание четырехмерного шума, который является незаменимым для анимационных аффектов.

Паллистер [3] выдвинул идею объединения четырех текстур, созданных с помощью алгоритма шума Перлина, для создания облаков. Первая текстура называется первой октавой. Она используется для задания общей формы облаков. Вторая октава генерируется в меньшем масштабе, чем первая, смешивается с первой для придания большей детальности. Аналогично третья и четвертая октавы генерируются в еще меньшем масштабе, чтобы сделать результирующую текстуру еще более детальной. Для анимации текстуры Паллистер предложил непрерывно модифицировать текстуру, интерполируя ее со вторым набором текстур динамически генерируемого шума. Дубе [4] использовал аналогичный подход с восьмью октавами и добавил алгоритм трассировки лучей в шейдере, чтобы вычислить реалистичное освещение в каждом пикселе. Проблема здесь заключается в том, что такие шейдеры требуют слишком много вычислительных ресурсов. Такое представление облаков, таким образом, делает их недоступными для экранов с высоким разрешением.

Цели и задачи статьи.

Необходимо осуществить процесс создания реалистичных изображений облаков и неба, выполнить генерацию текстуры облаков с помощью алгоритма шума Перлина, реализовать цикл смены дня и ночи для использования в компьютерных играх. Должно быть обеспечено быстроедействие, достаточное для представления таких эффектов окружения в реальном времени.

Реализация алгоритма.

Для реализации анимированных облаков нами были сгенерированы 4 октавы шума Перлина на разных частотах и амплитудах. В общем случае количество октав влияет на реалистичность конечного изображения, но для создания облаков

четырёх октав вполне достаточно.

Каждая октава рассчитывается отдельно, а затем, сложив все эти функции, мы получим результирующую – в достаточной степени гладкую и хаотичную. Создавая текстуры шума несколько раз и объединяя затем в одну, мы добиваемся более реалистичного представления облаков. Однако после такой процедуры сглаживание шума все же требуется. Функция сглаживания возвращает цвет пикселя в зависимости от исходного цвета и цвета соседних пикселей.

Сгенерированную таким образом текстуру облаков необходимо поместить на сферу, иначе облака будут казаться бесконечно уходящими вдаль.

Как было сказано выше, всего генерируется 4 октавы шума. Анимация облаков, таким образом, осуществляется простой интерполяцией от одного шума к другому. Чаще всего функция интерполяции принимает 3 параметра. Первые два, a и b , – параметры, между которыми производится интерполяция, последний – некоторое число в диапазоне от 0 до 1, от которого будет зависеть результат интерполяции: в случае, если параметр равен 0, то возвращается a , если равен 1, то возвращается b , иначе возвращается некоторое значение между a и b . Нужно также учесть, что октавы шума Перлина должны быть анимированы с разной скоростью: октава с большей амплитудой шума должна анимироваться медленнее октавы с меньшей амплитудой шума.

Для представления неба используются 3D модель купола неба, которая размещается над виртуальным миром и всегда центрирована на пользователе. Рендеринг купола неба осуществляется с помощью трех текстур, смешиваемых между собой в зависимости от времени суток. Текстура с голубым градиентом используется для дневного неба, а текстура, представляющая звезды, и текстура с темно-синим градиентом – для ночного.

Сгенерированная текстура облаков накладывается на купол неба. В зависимости от плотности облаков, текстура будет сделана частично прозрачной, позволяя куполу неба быть видимым. Каждая из этих моделей рендерится с помощью

пиксельных шейдеров на языке Cg. Шейдер для облаков вызывается один раз за кадр для каждого пикселя.

Предусмотрены два контролируемых параметра, влияющих на выполнение шейдера: количество облаков и время суток. Время суток меняется перемещением соответствующего ползунка, который позволяет пройти полному суточному циклу. Для задания количества облаков используются соответствующие кнопки, всего предусмотрено 4 степени концентрации облаков.

Результаты работы представлены на рисунке 1.



Рисунок 1 – Результаты генерации облаков

Выводы и перспективы.

Реализация рассмотренного алгоритма процедурной генерации облаков позволяет создавать реалистичное изображение. Поскольку алгоритм реализован с использованием графического процессора, следует ожидать, что в результате достигнуто быстрое действие, достаточное для генерации

изображения облаков в реальном времени. Полученные результаты будут использованы для создания динамического игрового 3D окружения.

Список литературы

6. Perlin, K. Implementing Improved Perlin Noise. – GPU Gems: programming techniques for high-performance graphics and general-purpose computation (Matt Pharr ed), 2005 [электронный ресурс]

http://http.developer.nvidia.com/GPUGems/gpugems_ch05.html

7. Green, S. Implementing improved Perlin noise. – GPU Gems 2: programming techniques for high-performance graphics and general-purpose computation (Matt Pharr ed), 2005 [электронный ресурс]

http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter26.html

8. Pallister, K. Generating Procedural Clouds Using 3D Hardware. – Game Programming Gems 2 (Mark Deloura, ed.), Charles River Media, 2001, pp. 463-473.

9. Dube, J. “Realistic Cloud Rendering on Modern GPUs” in Game Programming Gems 5 (Kim Pallister, ed.), Charles River Media, 2005, pp. 499-505.

УДК 004.27

А.Ю. Плитка, А.А. Потопахин, О.А. Гудаев
Донецкий национальный технический университет

ПЯТОЕ ИЗМЕРЕНИЕ – ПРОГРАММИРОВАНИЕ

Аннотация.

Плитка А.Ю., Потопахин А.А., Гудаев О.А. Пятое измерение – программирование. Данная работа посвящена общим вопросам информатики и программирования. В ней рассматриваются две гипотезы, с помощью которых можно представить пятое измерение как программирование.

В нашей повседневной жизни мы обычно отбрасываем бесполезную информацию, и нам нет дела до существования пятого измерения. Хотя, астрофизики допускают, что двадцать шесть измерений масштабируются в пятом [1, 2]. Но для чего все эти перевоплощения, когда простое объяснение геометрии движения всего перегрузили разбуханием мембран и струн?

Например, мы знаем, что человеческий глаз, как регистратор информации, видит только два пространственных измерения, а третье вычисляется мозгом. Опыт перемещения тела в глубину комнаты подтверждается только данными тактильных сенсоров, а изображение глаз всё равно плоское.

Если представить себя программой, состоящей из электромагнитных волн, а не системой химических элементов, то в таком состоянии мы бы не смогли распознать пространственную координату глубины, так как никакие объекты не ограничивали бы наше движение. И если бы свет был не фотоном, а материальным носителем, то мы могли бы полететь со скоростью света в соседнюю звёздную систему.

Вопрос только в одном: программатор и исполнительный механизм должны вычислять быстрее скорости света. Модулировать световой сигнал – это работать на частоте, умножающей константу осцилляции света [3]. Подозрительно, что у скорости света есть предельное значение.

Оно должно быть бесконечным, если свет – фотон. Поэтому, согласимся на реон [4].

Давайте вспомним про анабиоз жаб, понижающих жизненные процессы при замерзании. Второй факт: чёрная дыра является мотором, раскручивающая Вселенную. А сейчас за ней замечена работа «печкой» выстреливающей газами для спекания вещества в звёзды [5]. Из этого можно сделать вывод, что черная дыра должна замедлять скорость света для программирования сигнала, а затем выстреливать программой в любом направлении.

Поэтому, необходимо найти исполнительный механизм для свето-представления:

- другие черные дыры. Это просто и логично;
- звезды. Чтобы быть ретрансляторами сигнала, как узел в сети Интернет, надо думать о реоне и его главном механизме – баллистической теории света [4];
- гигантская ловушка. Необычный вариант.

Рассмотрим последний вариант. В качестве «гигантской ловушки» подойдет планета с океаном. Огромная водяная линза может являться вычислительным полигоном для прогонки программ. Логично заметить, что яркий солнечный свет не доходит до дна океана, а рассеивается по пути. Представим, что у планеты нет атмосферы, и свет падал бы не на дно океана, а на огромное водяное облако пара – водяной газ.

Попытаемся понять гигантскую ловушку, как вычислительную машину. Универсальная вычислительная машина Тьюринга основана на двух этапах имитационного моделирования: изменяйся и остывай после работы, готовясь использовать предыдущее состояние, почти память на элементах задержки, для последующих преобразований.

Вращающаяся планета частично закрывается от света. Образуется тень на обратной стороне планеты, не обращенной к солнцу-ретранслятору программ. Предполагая, что ретранслятор – это лента машины Тьюринга, а пар – это вычислитель. Солнце выступает в роли ПЗУ.

В качестве реального примера такого представления накопителей могут быть бактерии. Китайские ученые научились

хранить информацию в колонии бактерий – потенциально такая живая память способна накапливать 90 Гб на 1 г живого веса [7]. Бактериальные клетки можно настроить так, чтобы они записывали и хранили в своей ДНК любую информацию о внешней среде, даже если самим бактериям эта информация не нужна.

В машине Тьюринга должно быть две ленты: одна с не модифицируемой программой, другая чистая лента для сохранения состояний. Дедукция подсказывает, что поможет альтернативный вариант описания организации вычислений: ДНК-вычислитель или квантовый компьютер. Для этих двух методов вычисления можно выявить похожие технологии: молекулярный вычислитель, клеточный автомат и клэйтронику.

Но догма, помноженная на традиции, делает архитектуру фон Неймана костюмом, одетым на каждого игрока честной компании. Они не просто скованы. Они лежат на игровом планетарном поле. Как это случилось?

Очень просто: у игроков нет чёткого алгоритма процесса поведения. Даже нет логической модели организации вычислений. Поэтому, нельзя приступить к теоретическому осквернению этих товарищей и практическому тестированию на виртуальных симуляторах. Новичка в студию – «Таргетированная программа». Может быть воинственную букву «Т» надо отбросить и получится «золотой» мальчик Электрум : аргетированный цепной код-капсула. Тогда, как приготовить блюдо из первичного бульона?

Разбираем на запчасти машину Тьюринга следующим образом. Во-первых, делаем ленту из программы. Во-вторых, делаем вторую ленту с формальным описанием аппаратной реализации применения части кода с первой ленты. В-третьих, добавляем третью чистую ленту для записи промежуточных данных.

Конструируем альтернативную машину. Разрезаем ленты на части, по порядку, формируя цепной код. Помещаем порцию цепного кода в капсулу. Капсулу таргетируем: добавляем старт- и стоп-кадр и маркируем уникальным именем. Правильней будет

назвать такие информационные части капсулы четвёртой лентой. Четвёртая лента содержит секретный код: порядок следования и правила связывания капсул.

Эмулятор работы альтернативной машины очень прост: бросаем капсулы в ванночку; ждём, пока получится результат; неустанно следим под электронным микроскопом, возможно, что-то получится гениальное.

Конструктивные вопросы инженеринга аретированных программ:

1. Строгая очерёдность взаимодействия капсул. Можно ли что-то делать параллельно и возможно деантижадно? Антижадно – это когда вычисление «ленивое», т. е. отложенное на потом, а вот деантижадно – это опережая события, когда работа сделана впрок в параллельной реальности, несколькими способами, и отбирается только подходящий по случаю вариант, а остальные были избыточны.

2. Механизм обмена данными между капсулами должен учитывать состояние «спячки» капсулы, пока не дошёл её черёд вычислять.

3. Синхронизация. Надо ли передавать эстафету вычислений или динамически назначать ведущего? Как долго трусить ванночку до получения результата?

Если Вы раньше представляли дистрибутивную онтологическую систему знаний, реализуемую мультиагентной компьютерной системой, как движение шаров на бильярдном столе, то я Вас разочарую. Это похоже больше на наполнитель подушки из чистой овечьей шерсти. Например, лента ДНК, для компактности, скручивается в клубок, принимая шаровидную форму.

Вернёмся к планетарному масштабу пятого измерения. Как заметить в естественном сигнале аретированную программу? Надо найти капсулу.

Есть первая гипотеза R. Выявить капсулу можно, если обнаружить последовательность <белый шум, сложная форма, очень простая форма, белый шум>. Общий анализ сложной

формы исследован в работе [8]. Началом сенсорограммы будет сложная форма, а окончанием конец простой формы (см. рис. 1).

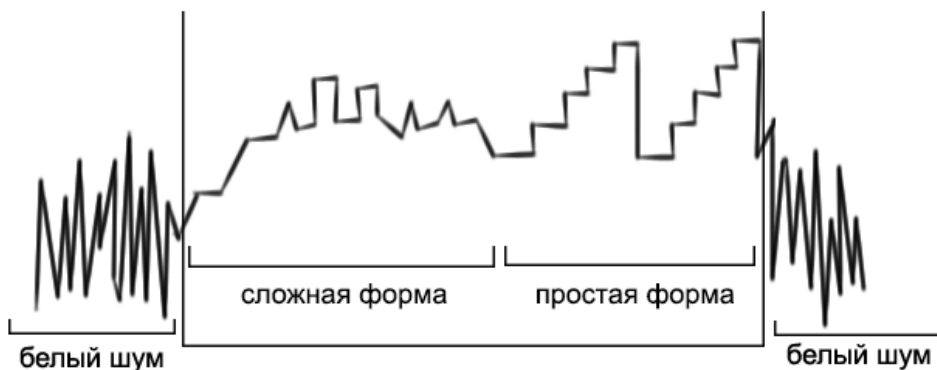


Рисунок 1 – Аргетированная программа

Тогда перевернутая гипотеза Р – это последовательность <белый шум, простая форма, сложная форма, белый шум>, которая движется в обратном направлении и для её интерпретации её надо развернуть.

Летит к нам свет, как аргетированная программа, переносимая вибрациями реонов. Возможно, когда-то в будущем мы пошлём своего виртуального аватара, от околочёрнодырочной орбиты, в виде света, к далёким звёздам для продолжения эволюции жизни в Галактике.

Выводы. В данной работе рассмотрено, как может выглядеть картина мира, основой которой является программирование. В образовательных целях её называем «пятым измерением». В работе было сформулировано две гипотезы, необходимые для поиска в естественных сигналах искусственных объектов – сценариев программирования.

Список литературы

1. Федосин, С. Г. Масштабное измерение как пятое измерение пространства-времени [Электронный ресурс] / С. Г.

Федосин. – Режим доступа: <http://serg.fedosin.ru/mi5.htm>. – Загл. с экрана. – (31.03.2015).

2. Масштабное измерение – Викизнание [Электронный ресурс]. – URL: http://www.wikiznanie.ru/ru-wz/index.php/Масштабное_измерение. – Загл. с экрана. – (01.04.2015).

3. Астафуров, В.И. Вывод и анализ уравнения, связывающего параметры фундаментальных взаимодействий со свойствами физического вакуума [Электронный ресурс] / В.И. Астафуров, А.М. Маренный. – Режим доступа: <http://matras.itep.ru/npd2k07/Konrer%202007/astafur.pdf>. – Загл. с экрана. – (01.04.2015).

4. Семиков, С. А. Структура света или самое тёмное дело в истории физики [Текст] / С. А. Семиков // Инженер. – 2008. – № 5. – С. 24–28; То же [Электронный ресурс]. – Режим доступа: <http://ritz-btr.narod.ru/svetoluch.html>. – Загл. с экрана. – (29.03.2015).

5. Ученые впервые наблюдали обеднение галактик из-за черных дыр: Космос: Наука и техника: Lenta.ru [Электронный ресурс]. – Режим доступа: <http://lenta.ru/news/2015/03/26/smbh/>. – Загл. с экрана. – (01.04.2015).

6. Гудаев, О. Иероглификация вселенского масштаба [Текст] / О. Гудаев // Наука и техника. – 2014. – № 7(98). – С. 7–10.

7. Живая память: Бактерии как носитель информации: Технологии: Popmech.ru [Электронный ресурс]. – Режим доступа: <http://www.popmech.ru/technologies/11255-zhivaya-pamyat-bakterii-kak-nositel-informatsii/>. – (26.04.2015).

8. Гудаев, О. А. Комбинаторика эквивалентных слов для проектирования лексикографических кодов расширенной реальности [Текст] / О. А. Гудаев // Искусственный интеллект. – 2014. – № 2. – С. 51–74. – ISSN 1561-5359.

УДК: 94(477):342.24

В. Л. Сверчков, Д. А. Завадский

Руководитель: Кривошеев Сергей Васильевич ассистент,
Завадская Татьяна Владимировна доцент,
Донецкий национальный технический университет
Кафедра компьютерной инженерии,

ОРГАНИЗАЦИЯ СИМУЛЯТОРА-ТРЕНАЖЁРА ИНЖЕНЕРНОЙ ТЕХНИКИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ.

Аннотация

Сверчков В. Л. Завадский Д. А. Организация симулятора-тренажёра инженерной техники с использованием технологии распределенных вычислений. Симуляторы-тренажёры инженерной техники используются для повышения квалификации, специалистов и способствует профилактике и предотвращению возникновения чрезвычайных ситуаций.

Развитие тяжёлой, химической и прочих опасных видов промышленности ведёт к усложнению технологий, повышению уровня квалификации специалистов и опасности производства, что влечёт за собой повышение степени возникновения техногенных аварий, обусловленных человеческим фактором. Основные возможные пути снижения влияния этого фактора следующие:

- Внедрение новых качественных методов подготовки персонала;



Рисунок -1. Hardware
составляющая тренажёра

- Глубокая роботизация производства.

Важность обучения людей, способных за короткое время решить поставленную перед ними задачу, является беспрецедентной и, порой, от

таких специалистов могут зависеть даже чьи-то жизни.

В связи с увеличением количества техногенных, экологических и природных катастроф, всё более становятся востребованы инженеры и специалисты, владеющие навыками работы на специализированной инженерной технике, в том числе на гусеничной платформе. Для управления данной техникой требуются высокие профессиональные качества водителей.

Одним из путей повышения профессиональных навыков водителей гусеничной инженерной техники является применение тренажёров.

Все симуляторы можно условно разделить на две большие группы – игровые и учебные. Две эти большие группы можно так разделить на несколько позиций.

Учебные симуляторы	Игровые симуляторы
<ul style="list-style-type: none"> • Авиационные симуляторы; • Симуляторы поездов; • Автосимуляторы; • Симуляторы инженерной техники; • Симуляторы плавающих средств. 	<ul style="list-style-type: none"> • Танковые симуляторы; • Космические симуляторы; • Автосимуляторы; • Другие.

Таблица 1. Разновидности симуляторов

На рынке современных тренажёрных систем присутствуют симуляторы компаний TRANSAS (морские суда), Caterpillar (тяжелая строительная техника, см Рис.4), Microsoft (авиационная и железнодорожная техника, см Рис.2). Программные продукты первых двух компаний являются полномасштабными симуляторами, с использованием специальных инженерных



Рисунок -2. Симулятор Microsoft Train Simulator

решений, однако они являются платными и могут быть получены только сертифицированными центрами. Тренажёры от корпорации Microsoft являются бесплатными. Для понимания сути симуляторов этого разработчика был произведён обзор хорошо зарекомендовавшего себя проекта Microsoft Flight Simulator.

Основными особенностями последней версии симулятора полётов от Microsoft являются:

- Разнообразие техники;
- Окружающий, «живущий» мир
- Различная погода;
- Высокодетализированные кабины пилотов, с возможностью взаимодействовать с каждым из элементов управления;

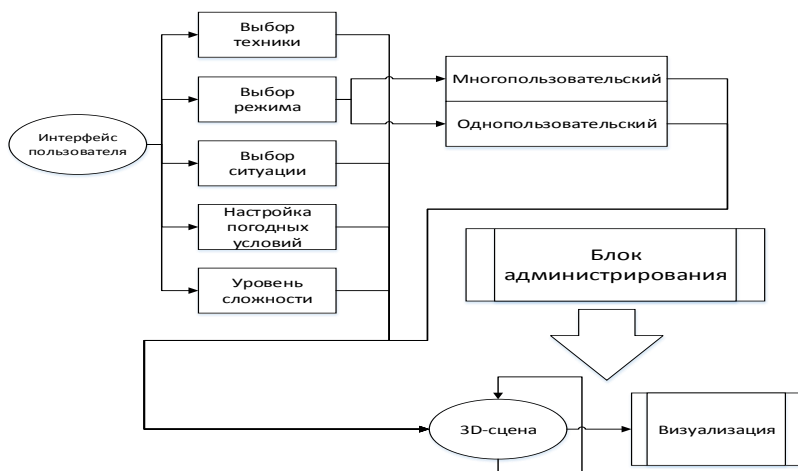


Рисунок -.3. Структура симулятора-тренажёра

Многие из описанных возможностей так же будут приняты во внимание, и в дальнейшем реализованы в разрабатываемом проекте. Структура подобного симулятора-тренажёра может выглядеть следующим образом (рис.3), которая включает в себя следующие модули, блоки:

- Блок интерфейса пользователя. В интерфейсе происходит глобальная настройка системы;



Рисунок-4. Симулятор от

условиями и сбор статистики во время и после процесса обучения.

- Блок организации одно- или многопользовательского режима обучения;
- Блок обработки графики. Построение и вывод картинки на графическое устройство;
- Блок администрирования.

Обеспечивает контроль над

Данный тренажёр может использоваться не только для подготовки водителей инженерной гусеничной техники, но и боевой техники, расположенной на гусеничном шасси[3].

Основными проблемами при реализации данного проекта является обеспечение уровня реалистичности, адекватности и соответствие масштабу реального времени. Реалистичность в процессе симуляции может быть достигнута за счёт того, что будут учтены все воздействующие физические силы, и особенности устройства управляемой гусеничной машины. Например: сила воздействия на грунт, корректность работы передач трансмиссии, особенности рельефа, правильность взаимодействия с текущим объектом, расчёт ускорения и т.д. Система реального времени, а именно реакция среды на объект и объекта на среду за конечное время, которое будет удовлетворять требования постоянного и своевременного ответа от системы пользователю является очень важным компонентом в симуляторе[1][2]. Данная система может быть реализована за счёт своевременного захвата, и как следствие, освобождения системных ресурсов, буферизации часто используемых данных и

привязки программных событий к моментам времени.

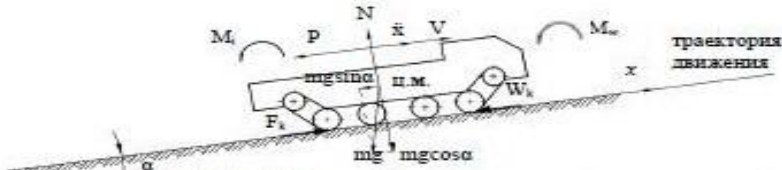


Рисунок-5. Математическая модель гусеничного транспортного средства.

В качестве повышения быстродействия разрабатываемого симулятора были рассмотрены два метода вычислений:

- Распределённые вычисления;
- Параллельные вычисления.

Другой путь повышения производительности тренажёра является использованием многоядерных или многопроцессорных систем (рис.6). Эти виды организации вычислений являются высокоэффективными в настоящее время и каждый из них имеет свои преимущества. Многие операционные системы поддерживают многопоточность, поэтому использование многоядерных процессоров позволяет ускорить работу компьютера даже в случае приложений, которые многопоточность не поддерживают.

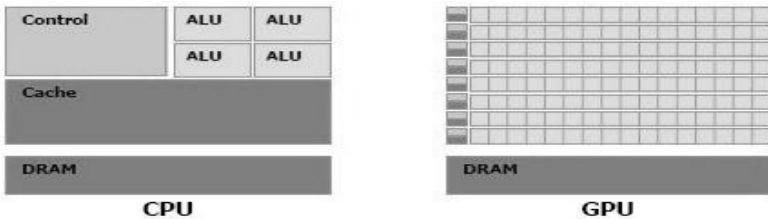


Рисунок-6. Сравнение архитектуры CPU и GPU.

Отдельного внимания заслуживает технология корпорации NVidia – CUDA, которая позволяет существенно увеличить вычислительную производительность благодаря совместному использованию CPU и GPU. Технология CUDA позволяет использовать вычислительные мощности видеокарты не только для графических вычислений, но и для различных научных вычислений. Многие научные и коммерческие организации

используют данную технологию, увеличивая скорость работы приложений, написанных с использованием данной технологии, как минимум в 10 раз (компании Numerix и CompatibL анонсировали поддержку CUDA в новом приложении, и достигли как раз такого прироста к производительности). Платформа параллельных вычислений CUDA обеспечивает набор расширений для языков C и C++, позволяющих выражать параллелизм данных и параллелизм задач на уровне мелких и крупных структурных единиц.

Применение тренажёров гусеничной инженерной техники способствует лучшему изучению специалистом особенностей своей профессиональной деятельности. Так как в ходе учебного процесса происходит симуляция чрезвычайных ситуаций, оператор тренажёрного комплекса получает полезные навыки работы, как и в одиночном порядке, так и в совокупности с командой. Обрётённые умения позволяют в значительной мере нивелировать влияние человеческого фактора в условиях чрезвычайной ситуации и, таким образом, проявлять высшую степень профессионализма в работе с инженерной техникой.

Список литературы:

1. *Хемди А. Таха*. Глава 18. Имитационное моделирование // Введение в исследование операций = Operations Research: An Introduction. — 7-е изд. — М.: «Вильямс», 2007. — С. 697-737.
2. *Строгалев В. П., Толкачева И. О.* Имитационное моделирование. — МГТУ им. Баумана, 2008. — С. 697.
3. *М. Барятинский*. Средние и основные танки зарубежных стран (Часть 2). — Москва: Моделист-конструктор, 2002. — 32 с. — (Бронекolleкция № 2 (41) / 2002). — 4500 экз.

УДК 681.322

Д.С. Сипаков, А.Ю. Харитонов

Донецкий национальный технический университет, г. Донецк
кафедра компьютерных систем мониторинга

E-mail: dimasipakov@gmail.com

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВРЕМЕНИ ИСПОЛНЕНИЯ МАТЕМАТИЧЕСКОГО РЕШЕНИЯ НА ПЭВМ И КОНТРОЛЛЕРЕ С FPGA-ПРОЦЕССОРОМ

Аннотация

Сипаков Д.С., Харитонов А.Ю. Сравнительный анализ времени исполнения математического решения на ПЭВМ и контроллере с FPGA-процессором. С помощью среды LabView выполнен математический расчет интеграла. Полученные решения запущены на выполнения на компьютере и контроллере FPGA с целью анализа времени исполнения.

Ключевые слова: контроллер, интеграл, FPGA, визуальное программирование.

Постановка проблемы.

В ходе выполнения лабораторного практикума по курсу «Автоматизация» была поставлена задача: исследовать возможности FPGA-процессоров на контроллерах и языка программирования LabVIEW. Для решения этой проблемы было принято решение произвести сравнительный анализ по времени выполнения одного и того же математического расчета на компьютере и на контроллере с FPGA-процессором.

Цель статьи

Произвести сравнительный анализ времени выполнения одного математического решения на компьютере и контроллере, с целью определения возможностей контроллера с FPGA-процессором.

Решение задач и результаты исследований.

Программируемая логическая интегральная схема (ПЛИС) — электронный компонент, используемый для создания цифровых

интегральных схем. В отличие от обычных цифровых микросхем, логика работы ПЛИС не определяется при изготовлении, а задаётся посредством программирования (проектирования). Для программирования используются программаторы и отладочные среды, позволяющие задать желаемую структуру цифрового устройства в виде принципиальной электрической схемы или программы на специальных языках описания аппаратуры. Альтернативой ПЛИС являются: программируемые логические контроллеры (ПЛК), базовые матричные кристаллы (БМК), требующие заводского производственного процесса для программирования; ASIC — специализированные заказные большие интегральные схемы (БИС), которые при мелкосерийном и единичном производстве существенно дороже; специализированные компьютеры, процессоры (например, цифровой сигнальный процессор) или микроконтроллеры, которые из-за программного способа реализации алгоритмов в работе медленнее ПЛИС [1].

ПЛИС используют в приложениях, где необходимо большое количество портов ввода-вывода, цифровая обработка сигнала, высокоскоростная передача или обработка данных. Одним из типов ПЛИС является FPGA.

FPGA (Field-Programmable Gate Array) — программируемая пользователем вентильная матрица (ППВМ). содержат блоки умножения-суммирования, которые широко применяются при обработке сигналов (DSP), а также логические элементы (как правило, на базе таблиц перекодировки — таблиц истинности) и их блоки коммутации. FPGA обычно используются для обработки сигналов, имеют больше логических элементов и более гибкую архитектуру, чем CPLD. Это полупроводниковое устройство, которое может быть сконфигурировано производителем или разработчиком после изготовления. Устройство программируется путём изменения логики работы принципиальной схемы, например, с помощью исходного кода на языке проектирования, на котором можно описать эту логику работы микросхемы.

Логический блок классической ППВМ состоит из таблицы истинности на 4 входа и триггера.

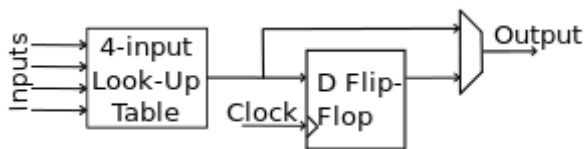


Рисунок 1 — Типичный логический блок

Логический блок имеет таблицу истинности на 4 входа и вход синхронизации (clock). Выход блока только один, это может быть регистровая или нерегистровая выходная таблица истинности. Поскольку сигналы синхронизации в коммерческих ППВМ (а часто и другие сигналы, распараллеливающиеся на большое количество входов — high-fanout signals) трассируются особым образом специальными трассировочными, управление этими сигналами делается отдельно [2].

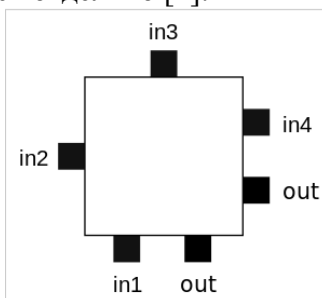


Рисунок 2 — Расположение контактов логического блока

В месте пересечения вертикальных и горизонтальных каналов создаются переключательные блоки. При такой архитектуре для каждого проводника, входящего в переключательный блок, существуют три программируемых переключателя, которые позволяют ему подключаться к трём другим проводникам в смежных сегментах канала.

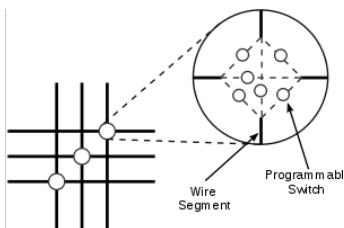


Рисунок 3 — Топология переключательного блока

Рассмотрим некоторые виды контроллеров с FPGA. Один из них NI cRIO-9072. Он имеет 8-слотное шасси и процессор мощностью 266 МГц. Имеет последовательный порт RS232 для подключения периферийных устройств, а так же T-Enthernet порт.



Рисунок 4 — NI cRIO-9072

Еще одним из подобных контроллеров является NI 9870. Он имеет 4 последовательных порта RS232. Различные скорости передачи в бодах от 14 бод до 921,6 кбод. Информационные биты: 5, 6, 7, 8; стоповые биты: 1, 1,5, 2; управление потоком: XON/OFF, RTS/CTS. От 8 до 28 В постоянного тока от внешнего источника питания; кабель PC-MF4-PT включен в комплект поставки. Рабочий диапазон температур от -40 до 70 °С.



Рисунок 5 — NI 9870

Последним из рассматриваемых контроллеров является NI 9211. Имеет 4 термодатчики или аналоговые входы на ± 80 мВ, 24-битная резолуция, рабочий диапазон температур от -40 до 70 °С.



Рисунок 6 — NI 9211

Лабораторные работы проводились на контроллере NI cRIO-9072(см.рис.1). Для решения поставленной задачи изначально были реализованы два численных метода решения интегрального выражения. В среде LabView были выполнены программы, реализующие метод прямоугольников и метод трапеций, которые вычисляют приблизительно значение одной и той же подынтегральной функции.

После того, как необходимые программы были реализованы, они были запущены на выполнение на компьютере, время выполнения каждого расчета замерялось в миллисекундах.

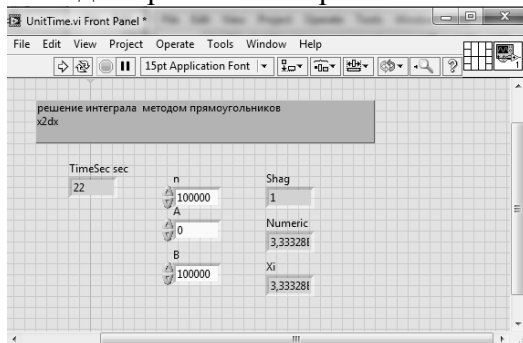


Рисунок 7 — Результат выполнения расчета метода прямоугольников на компьютере

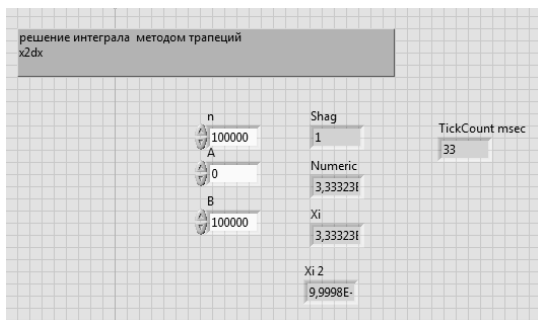


Рисунок 8 — Результат выполнения расчета метода трапеций на компьютере

Затем эти же программы были запущены на выполнение на контроллере NI cRio 9072, на нем время выполнения замерялось в наносекундах и тактах работы процессора.

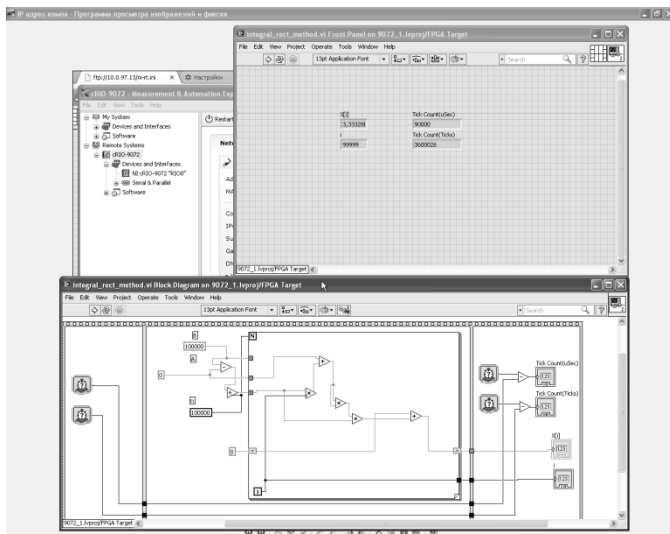


Рисунок 9 — Результат выполнения расчета метода прямоугольников на контроллере

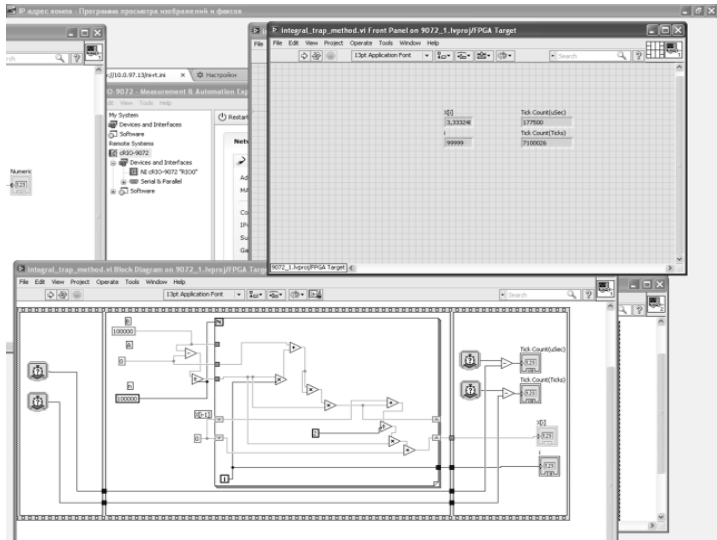


Рисунок 10 — Результат выполнения расчета метода трапеций на контроллере

По полученным результатам, которые отражены на рисунках 7,8,9,10 была составлена таблица, которая отображает время, затрачиваемое на выполнение расчетов на компьютере и на контроллере NI cRio 9072. Более того как можно заметить реализация программ математических расчетов на языке LabView на ПЭВМ ничем не отличается от реализации на контроллере. Код на LabView сохраняется на контроллере почти в том же виде, в котором он был создан пользователем.

Таблица 1 — Время выполнения расчетов

Выполняемый расчет	Время на компьютере, сек	Время на NI cRio 9072, сек
Метод прямоугольников	0.022	0.00009
Метод трапеций	0,033	0.0001775

Исходя из результатов таблицы, можно сделать определенные выводы. Контроллер на базе FPGA-процессора выполняет сложные математические вычисления гораздо

быстрее, чем компьютер, это происходит за счет того, что операция на FPGA-процессоре выполняется полностью за один такт работы в отличие от обычных процессоров, которые встроены в компьютеры.

Выводы

В ходе проведения анализа и выполнения практических работ, было установлено, что контроллеры с FPGA-процессором обладают большим быстродействием при расчете сложных математических вычислений и расчетов, чем обычные процессоры. Более того FPGA-процессоры могут программироваться непосредственно пользователем для решения определенного типа задач.

Контроллеры на базе FPGA-процессоров могут быть использованы в распределенных системах, для выполнения параллельных вычислений, гарантируя быстрое выполнение операций. Более того FPGA могут применяться как ускорители универсальных процессоров в суперкомпьютерах (Cray — XD1, SGI — проект RASC).

Список литературы

1. Википедия [электронный ресурс] — Режим доступа: <https://ru.wikipedia.org/wiki/ПЛИС#GAL>
2. Википедия [электронный ресурс] — Режим доступа: https://ru.wikipedia.org/wiki/Программируемая_пользователем_вентильная_матрица
3. National Instruments [электронный ресурс]—Режим доступа: <http://sine.ni.com/nips/cds/view/p/lang/ru/nid/208787>
4. National Instruments [электронный ресурс]—Режим доступа: <http://sine.ni.com/nips/cds/view/p/lang/ru/nid/204259>
5. National Instruments [электронный ресурс]—Режим доступа: <http://sine.ni.com/nips/cds/view/p/lang/ru/nid/203963>
6. Е. Баран LabView FPGA. Реконфигурируемые измерительные и управляющие системы — М.: ДМК Пресс, 2014, 360.
7. National Instruments Учебный курс LabVIEW и CompactRIO: основы разработки приложений, 2009, 305стр.

Н.Н. Шермет

Донецкий национальный технический университет,
кафедра программного обеспечения интеллектуальных систем

ИССЛЕДОВАНИЕ МЕТОДА АКТИВНЫХ КОНТУРОВ И ЕГО МОДИФИКАЦИЙ

Аннотация

Шермет Н.Н. Исследование метода активных контуров и его модификаций. Выполнен анализ основных методов распознавания образов на изображении. Выполнен анализ метода активных контуров. Выполнен анализ существующих модификаций метод активных контуров. Предложены варианты модификации метода активных контуров.

Ключевые слова: методы обработки изображений, выделение границ объектов, образ, распознавание изображений, контурный анализ, активные контуры, адаптивное выделение.

Постановка проблемы: В последнее время распознавание образов находит все большее применение. Распознавание печатного и рукописного текста, различных объектов или движений упрощает взаимодействие компьютера и человека, а так же является подспорьем для создания искусственного интеллекта. Создавая устройства и приложения, выполняющие функции распознавания различных объектов, позволяет заменить человека механическим автоматом, где это может потребоваться. И если качество работы человека зависит от большого количество факторов, таких как внимание, усталость, квалификация, то механическая система будет работать одинаково хорошо и будет нуждаться только в небольшом контроле, на случай необходимости введения коррекции в работу. Так же, наибольшим преимуществом автоматизированных систем над человеком, является скорость выполнения ими работы, а так как техника развивается стремительными темпами и вычислительные

мощности растут с каждым годом – этот разрыв в скорости становится все больше.

Распознавание образов находит применения во многих сферах. Это могут быть: взаимодействие человека и компьютера (распознавание рукописного текста, распознавание лиц и детекторы движений), слежение за объектами (видеонаблюдение, трекинг передвижения целей), распознавание органов и их повреждений в медицине, распознавание окружающего пространства и навигация в кибернетике и т.д.

На сегодняшний день существует множество различных способов распознавания контуров, но для всех существует два ограничителя по качеству работы: скорость работы и качество нахождения контура. Для статических изображений, чаще всего, применяют «тяжелые» алгоритмы, наподобие нейросетей. Даже если не учитывать скорость работы нейросетей, у них есть зависимость от обучения, которое также занимает немало времени и стороннего контроля. В случае поиска динамических контуров, например в видеопотоке, задача усложняется в разы и медленные обучаемые алгоритмы не подходят для их качественного обнаружения.

Так как при распознавании статических образов, на текущий момент, проблем не возникает, то проблематичной остается распознавание динамических образов. Проблемами распознавания динамических образов являются: скорость определения месторасположения объекта, трекинг передвижения, учет возможности изменения формы контура во время его отслеживания.

Анализ литературы: Проведен анализ существующих реализаций контурного анализа и способы реализации метода активных контуров [1][2][3].

Цель статьи: провести анализ алгоритма поиска образов методом активных контуров, определить его достоинства и недостатки по сравнению с другими методами, проанализировать его модификации.

Постановка задачи исследования. На первой стадии исследования необходимо определить способы предобработки получаемого изображения A (или же серии изображений A_i) методом F так, что бы получаемое бинаризованное изображение B содержало в себе бинарный контур объекта с минимальным количеством шумов. На второй стадии требуется задать (или же определить) стартовое место расположения контура и обработать изображение B_i так, что бы на выходе работы метода, получить координаты точек контура.

Решение задач и результаты исследований.

Для решения задачи поиска контура объекта, были проанализированы несколько методов нахождения контуров, в следствии поделенные на группы:

- 1) Методы пороговой обработки. Группа простейших алгоритмов, суть которых заключается в том, что бы отнести точки изображения к определенной группе на основании их яркости. К ним относятся пороговая сегментация (Рисунок 1.1) и фильтр водораздела. В сегментации методом водораздела рассматривается абсолютная величина градиента изображения как топографической поверхности.

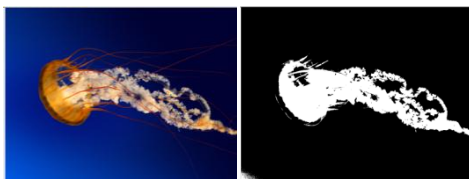


Рисунок 1 – Пороговая сегментация

- 2) Алгоритмы получения вторых производных изображения. Суть данных алгоритмов заключается в поиске всех перепадов яркости на изображении (его градиента). К таким алгоритмам относятся операторы Собеля (рисунок 1.2), Лапласа, Преввита, Кенни и тд. В

ходе работы оператора, над каждым пикселем изображения происходит операция свертки по заданной маске, соответствующей оператору.



Рисунок 2 – Оператор Собеля

- 3) Цепное кодирование контура. К таким способам выделения контура относятся цепной код Фримена, алгоритма «Жука» и тд.. Цепной код Фримена применяется для представления границы в виде последовательности отрезков прямых линий определённой длины и направления. В основе этого представления лежит 4- или 8- связная решётка.
- 4) К отдельной группе методов контурного анализа можно отнести метод активных контуров. Суть метода (рисунок 1.4) заключается в расчете отклонения исходного контура, представленного последовательностью точек расположенных в двумерном пространстве, от бинаризованного контура объекта на самом изображении. Работа метода заключается в том, что для каждой точки изображения, близко лежащей к точкам контура v_i , рассчитывается разность между «распирающей» и «сдавливающей» силами и в последующем перемещении точки v_i , относительно которой рассчитывались эти силы, в точку изображения с минимальной разностью

он тяжел в вычислении относительно других способов выделения границ, а так же то, что требуется начальное приближение к объекту. Данные проблемы могут быть решены при помощи различных адаптивных методов выделения контуров, которые позволяют автоматически задавать начальное приближение. К ним относятся модификации с использованием приближения по шаблону и полному покрытию изображения.

Выводы. Проведен анализ методов выделения контуров на изображении. Проанализирован алгоритм работы метода активных контуров. Произведена оценка положительных и отрицательных свойств метода, а так же его различные модификации. Проанализированы недостатки метода и способы их устранения.

Список литературы

1. Active contour model / Интернет-ресурс. - Режим доступа: [www/
http://en.wikipedia.org/wiki/Active_contour_model](http://www.wikipedia.org/wiki/Active_contour_model) - URL: [URL: http://en.wikipedia.org/wiki/Active_contour_model](http://www.wikipedia.org/wiki/Active_contour_model) - Загл. с экрана.
2. Модификация алгоритма активных контуров для решения задачи интерактивной сегментации растровых изображений дефектов металлический отливок / Интернет-ресурс. - Режим доступа: [www/ URL: http://www.science-education.ru/30-1128](http://www.science-education.ru/30-1128) - Загл. с экрана.
3. Ярославский Л.П. Введение в цифровую обработку изображений. / Ярославский Л.П. – М.: Сов. радио, 1979. – 312 с.



**КОМПЬЮТЕРНА
ИНЖЕНЕРИЯ**

СЕКЦИЯ 6

УДК 681.3

Гимадеев К.Д., Теплинский С.В., Франжев К.Е.
Донецкий национальный технический университет
Кафедра компьютерной инженерии
E-mail:kiria-b2@mail.ru
disep@mail.ru

АНАЛИЗ И ИССЛЕДОВАНИЕ ТАЙМЕРОВ В WINDOWS

Гимадеев К.Д., Теплинский С.В., Франжев К.Е. Анализ и исследование таймеров в Windows и C#. В статье рассматривается вопрос использования таймеров операционной системы Windows в языке программирования C#. Исследованы потери сообщений о тике таймера и сделаны предположения о некоторых аспектах работы таймера. Также исследована работа мультимедиа таймера.

Введение

Таймер в Windows является устройством ввода информации, которое периодически извещает приложение о том, что истек заданный интервал времени. В системе Windows таймер посылает сообщение WM_TIMER, которое можно обработать в программе, тем самым выполняя заданные действия с периодичностью, указанной при инициализации таймера. Интервал таймера может находиться в пределах (теоретически) от 1 до 4294967295 миллисекунд, что составляет около 50 дней [2].

В библиотеке .NET Framework представлены 3 класса с таймерами в различных пространствах имён. Это System.Timers.Timer, System.Threading.Timer, System.Windows.Forms.Timer [3]. Данные классы реализуют системные таймеры, которые через заданный промежуток времени посылают системе синхронные сообщения, с приходом которых вызывается обработчик таймера.

Работа таймеров

Все 3 класса устроены одинаково, но предназначены для разных нужд:


```

Код С#:
using System.Threading;
Timer timer1 = new Timer(delegate{ tics++; }, null, 0, 500);
Timer timer2 = new Timer(delegate{
Console.WriteLine(tics.ToString()); tics = 0;},null, 0, 10000);

```

- `System.Threading.Timer` – самый простой класс таймера, используется для выполнения команд через заданный промежуток времени.

- `System.Timers.Timer` – таймер оптимизированный для работы в многопоточных приложениях или приложениях, имеющих асинхронные функции

- `System.Windows.Forms.Timer` – простой таймер, работающий в платформе Windows Forms.

Пример инициализации таймера `System.Threading.Timer` в листинге 1.

Листинг 1 – инициализация таймера.

Данная программа выводит в консоль через каждые 10 секунд количество пришедших сообщений таймера `timer1`. На интервале в 10000 мс, который отсчитывается таймером `timer2`, имеем такие данные о количестве тиков таймера `timer1`, приведённые в таблицах 1,2,3.

Таблица 1 – данные о потерях сообщений таймера `timer1` класса `System.Threading.Timer` на интервале 10с (10000мс)

Интервал таймера <code>timer1</code>	500мс	100мс	50мс	10мс	1мс
Количество полученных (ожидаемых) сообщений	20(20)	92(100)	160(200)	637(1000)	640(10000)
Потери сообщений в процентах	0%	8%	20%	36%	93,6%

Таблица 2 – данные о потерях сообщений таймера timer1 класса System.Timers.Timer на интервале 10с (10000мс)

Интервал таймера timer1	500мс	100мс	50мс	10мс	1мс
Количество полученных (ожидаемых) сообщений	19 (20)	91(100)	160(200)	637(1000)	639(10000)
Потери сообщений в процентах	5%	9%	20%	36%	93,6%

Таблица 3 – данные о потерях сообщений таймера timer1 класса System.Windows.Forms.Timer на интервале 10с (10000мс)

Интервал таймера timer1	500мс	100мс	50мс	10мс	1мс
Количество полученных (ожидаемых) сообщений	20(20)	92(100)	160(200)	637(1000)	640(10000)
Потери сообщений в процентах	0%	8%	20%	36%	93,6%

Проанализировав данные таймеры, мы пришли к выводу, что системные таймеры в различных пространствах имен C# работают одинаково. Более того, эксперименты с таймером WinAPI дали такие же результаты.

В литературе [1,2] приводятся данные о том, что сообщения системного таймера приходят с частотой 18,21 раза в секунду, т.е. минимальный интервал составляет 55мс. Наши эксперименты показывают, что интервал таймера можно устанавливать с дискретностью 1мс. Однако, как видно из таблиц, интервал менее

10 мс не влияет на количество полученных сообщений WM_TIMER.

Можно предположить, что такая потеря сообщений таймера может быть вызвана задержками на обработку сообщения таймера или другими причинами. Если в очереди сообщений появляются два и более сообщений от одного таймера, то они объединяются в одно. Вероятнее всего, другой причиной является регенерация памяти, которая запускается с интервалом не более 64мс [5]. В некоторых источниках указано, что частота интервала меньше 10 мс может отрицательно сказываться на работоспособности системы, поэтому в реализации таймера может стоять ограничение в 10мс.

Мультимедиа таймер

Для повышения быстродействия нами был использован мультимедиа таймер из библиотеки функций WinAPI. Нами были проведены тесты, которые показывают, что таймер, работающий на частоте 1кГц, обеспечивает достаточную скорость работы и минимальную потерю сообщений WM_TIMER при работе.

Таблица 4 - данные о потерях сообщений мультимедийного таймера

Интервал таймера	500мс	100мс	50мс	10мс	1мс
Количество полученных (ожидаемых) сообщений	20 (20)	100 (100)	200(200)	1000(1000)	10000 (10000)
Потери сообщений в процентах	0%	0%	0%	0%	0%

В стандартных средствах библиотеки .NET Framework нет мультимедиа таймера, поэтому мы импортировали некоторые функции из стандартной библиотеки WinAPI как показано в листинге 2.

Листинг 2 – импорт библиотек для подключения мультимедиа таймеа в C#

Код C#:

```
[DllImport("winmm.dll")]
private static extern int timeSetEvent(int delay, int resolution,
TimerEventHandler handler, IntPtr user, int eventType);
[DllImport("winmm.dll")]
private static extern int timeKillEvent(int id);
[DllImport("winmm.dll")]
private static extern int timeBeginPeriod(int msec);
[DllImport("winmm.dll")]
private static extern int timeEndPeriod(int msec);
```

При работе таймера, длительностью 10000 мс он посылает 10000 сообщений Windows которые успешно обрабатываются и не теряются с системной очереди сообщений.

В основе всех системных таймеров лежит аппаратный программируемый таймер, максимальная частота которого от 1МГц и выше. Он управляет системными прерываниями и сообщениями WM_PAINT. Можно предположить, что мультимедиа таймер отправляет асинхронные сообщения, которые не становятся в очередь и обрабатываются моментально, что позволяет достичь такой высокой частоты работы приложения. Однако это может сказываться на производительности системы в целом, что в свою очередь на современных компьютерах практически не заметно.

Для большинства программ, в которых используется таймер, такая высокая частота работы таймера не требуется. Однако такая частота может потребоваться при написании специализированных программ (работа с аудио/видео, обновление окна для плавной анимации). В таких случаях программист должен сам выбирать между обычным таймером и мультимедиа таймером, в зависимости от требуемой частоты работы таймера.

Необходимо учитывать тот факт, что при выполнении операций, на выполнение которых затрачивается времени больше, чем интервал тика таймера, могут случаться сбои в работе

программы. Это может быть связано с потерей сообщений мультимедиа таймера. Во избежание сбоев, программист должен учитывать время, необходимое на выполнение операций, исполняемых по тикку таймера.

Как видно на рисунке 1, при выполнении ресурсоёмкой операции в программе возникают ошибки.



```
file:///C:/Users/Карим/Desktop/Учеба/2-й курс/
11896
28556
1
1
-
```

Рисунок 1 – работа таймера с ресурсоёмкой операцией.

Выводы

В ходе работы были исследованы различные системные таймеры, предоставленные в библиотеке .NET Framework, а так же мультимедиа таймер. Написаны программы для исследования таймеров в консоли и проекте WinForm.

Рассмотрен вариант подключения мультимедиа таймера для операций, требующих высокой частоты работы таймера.

Список литературы

[1] Финогенов К.Г. Win32. Основы программирования. – 2-е изд., испр. и дополн. – М.: ДИАЛОГ – МИФИ, 2006. – 416с.

[2] Петзолд Ч. Программирование для Windows® 95 в двух томах

[3] Ресурс MSDN. / Интернет-ресурс. - Режим доступа : URL: <http://msdn.microsoft.com>

[4] Форум начинающих и профессиональных программистов / Интернет-ресурс. - Режим доступа : URL: <http://cyberforum.ru>

[5] Настройка подсистемы памяти в BIOS SETUP / Интернет-ресурс. - Режим доступа : URL: <http://www.ixbt.com/mainboard/memorytune-viabios.shtml>

УДК 004.052.4

Мохначёв В.В., Елагин И.А., Теплинский С.В.
Донецкий национальный технический университет
кафедра компьютерной инженерии
E-mail: vlad230596@mail.ru, progressiff@gmail.com

**ОПИСАНИЕ И ОБРАБОТКА СИСТЕМНЫХ И
ПОЛЬЗОВАТЕЛЬСКИХ ОШИБОК**

Мохначёв В.В., Елагин И.А., Теплинский С.В. Описание и обработка системных и пользовательских ошибок. В статье рассматривается вопрос обработки ошибок, возникающих при выполнении функций windows. Определен основной способ описания собственных ошибок и использование его при программировании с помощью подключения файла-ресурса и динамической библиотеки.

Введение

Безусловно, обработка ошибок является одним из самых важных процессов разработки приложения, так как программа должна быть понятной пользователю. Пользовательские ошибки возникают повсеместно: ввод неправильных данных, работа не в нужном режиме и т.д. Качественная программа должна определять возникновение ошибки, предоставлять информацию о ней пользователю, и по возможности предлагать варианты решения проблемы. Актуальность данной работы состоит в том, что не смотря на важность данной проблемы, описание её решения встречается редко, к тому же не в полной мере.

Обработка ошибок в функциях Windows

Функция Windows при вызове проверяет переданные ей параметры и начинает выполнять свою работу. Если, например, программист передал недопустимый параметр функция возвращает значение, которое свидетельствует об ошибке. За каждой ошибкой закреплено 32-битный код. Чтобы получить код ошибки используется функция DWORD GetLastError (). Ее вызывают сразу после выполнения функции Windows. Все

системные ошибки Windows описаны в файле winerror.h (обычно в литературе, например, «Рихтер Дж. Windows для профессионалов» [2]), рекомендуется использовать именно их).

Анализируя файл winerror.h можно увидеть, что в нем есть две категории ошибок. Первая категория – это коды WIN32 – ошибок, представленные в виде десятичного числа, например:

```
#define ERROR_FILE_NOT_FOUND 2L
```

L задает тип long, то есть 32 – х битное значение для компилятора MS Studio 2010.

Вторая категория – это коды ошибок, которые возвращают COM функции, например:

```
#define E_OUTOFMEMORY  
_HRESULT_TYPEDEF_(0x8007000EL)
```

Ошибки из этой категории представлены в виде шестнадцатиричных констант, которые имеют тип HRESULT. Этот тип является одним из средств контроля ошибок в COM и представляет собой 32 – битное число, в котором кодируется результат операции. Нужно отметить, что те же ошибки Win32 также переводятся в HRESULT при помощи специального макроса HRESULT_FROM_WIN32. Значения двоичного кода этих ошибок разбито на поля как показано в таблице 1.

Таблица 1. Представление кода ошибки в 32 – х битном значении

Биты	31	30 -27	26 - 16	15 - 0
Содержимое	Код стпени «тяжести» (severity)	Зарезервированные значения	Код подсистемы (facility code)	Код самой ошибки
Значение	0 – успех 1 - ошибка	Определяются Microsoft	Определен Microsoft	Определяется Microsoft или пользователем

Описание собственных ошибок и использование их при программировании.

В файле winerror.h описано огромное количество ошибок. Но часто возникает ситуация, когда программисту необходимо самому описать ошибку или интерпритировать по-другому ошибку, описанную в winerror.h. Для этого в Windows имеется мощное средство создания собственных сообщений Message

Compiler или MC[3]. Это средство используется для компиляции файлов с расширением .mc. В данном файле определяются сообщения.

Содержимое файла можно разделить на две части: секция - заголовок и секция определений сообщений.

В заголовке определяются имена и идентификаторы языка для определения сообщений в теле файла. В таблице 2 определены следующие значения заголовка

Таблица 2. Синтаксические определения для описания заголовка

Значение	Описание
MessageIdTypedef= <i>type</i>	Указывается тип кода сообщения
SeverityNames=(<i>name=number[:name]</i>)	Указывается важность сообщения
FacilityNames=(<i>name=number[:name]</i>)	Указывается тип подсистемы
LanguageNames=(<i>name=number:filename</i>)	Указывается язык, на котором будут определены сообщения
OutputBase= <i>number</i>	Указывается в какой системе счисления будет определен код ошибки в десятичной или шестнадцатичной

На рисунке 1 представлен пример описания заголовка файла сообщений.


```

; // язык сообщений
LanguageNames=(Russian=0x419:MSG_RUS)
LanguageNames=(English=0x409:MSG_ENG)
; // определение категорий сообщений
MessageIdTypedef=DWORD

SeverityNames=(Success=0:STATUS_SEVERITY_SUCCESS
Informational=1:STATUS_SEVERITY_INFORMATIONAL
Warning=2:STATUS_SEVERITY_WARNING
Error=3:STATUS_SEVERITY_ERROR
)

FacilityNames=( System=0x0FF
Application=0xFF
)

```

Рисунок - 1. Пример описания заголовка файла сообщений

В секции определения сообщений определяются сами сообщения на заданных языках и их вид. В таблице 3 определены следующие значения секции сообщений.

Таблица 3. Синтаксические определения для описания секции сообщений

Значение	Описание
MessageId=[<i>number</i> + <i>number</i>]	Указывается идентификатор сообщения в шест
Severity= <i>name</i>	Указывается описанное ранее значение в заголовке
Facility= <i>name</i>	Указывается описанное ранее значение в заголовке
SymbolicName= <i>name</i>	Макроопределение сообщения
Language= <i>name</i>	Указывается описанное ранее значение в заголовке
<i>message text</i>	Записывается текст на выбранном языке

На рисунке 2 представлен пример описания секции сообщений файла сообщений.

```

MessageId=0x1
Severity=Error
Facility=Runtime
SymbolicName=EMPTY_CMD_LINE_ERROR
Language=English
Command Line is empty.
.
Language=Russian
Недостаточно аргументов командной строки.
.

```

Рисунок - 2. Пример описания секции сообщений файла сообщений

Созданный заполненный файл необходимо скомпилировать. Для этого очень удобно использовать командную строку компилятора Visual Studio. Все, что нужно это добавить заданный файл, в командной строке дать команду `mc -A "%(FullPath)"`, где `mc` – это компилятор сообщений, флаг `-A` указывает таблицу кодировки ASCII, `FullPath` дает полный путь к компилируемому файлу, и указать выходы компиляции `%(Filename).rc`; `%(Filename).h` в виде файла ресурса и заголовочного файла. На рисунке 3 представлен пример описанных действий.

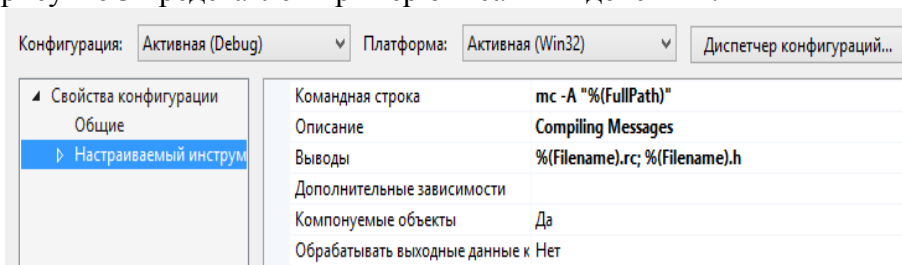


Рисунок - 3. Командная строка для компиляции файла сообщений

В результате компиляции получены заголовочный файл, подобный файлу `winerror.h` с описанием и идентификаторами в шестнадцатиричной системе счисления ошибок, структура идентификатора ошибки показана в таблице 4, бинарные файлы с

текстом ошибок на заданных языках, файл ресурс с определенными в нем полученными бинарными файлами и идентификаторами используемых языков. Заголовочный файл и файл ресурс подключаются к проекту и теперь можно использовать описанные ошибки.

Таблица 4. Представление кода описанной ошибки в 32 – х битном значении

Биты	31 - 30	29	28	27 - 16	15 - 0
Содержимое	Код степени «тяжести» (severity)	Кем определен – Microsoft или пользователем	Зарезервирован	Код подсистемы (facility code)	Код самой ошибки
Значение	00 – успех 01 – информация 10 – предупреждение 11 – ошибка	0 – Microsoft 1 - пользователь	Всегда ноль	Определен Microsoft	Определяется Microsoft или пользователем

Рассмотрим два способа подключения файла с ошибками к программе:

1. Встраивание ресурсного файла(*.res), полученного способом, описанным выше.
2. Создание динамической библиотеки, которая будет хранить описание ошибок.

Первый способ более простой в использовании, а также лучше портируемый, так как описание ошибок будет «вшито» в исполняемый файл. Минусы данного метода в том, что увеличивается размер исполняемого файла, и при использовании несколькими приложениями одних и тех же ошибок, нужно дублировать описание ошибок в каждом файле.

Второй способ позволяет создать библиотеку, которая может быть использована различными приложениями, а храниться она будет только в одном месте. Если есть необходимость создать динамическую библиотеку для описания архитектуры приложения, целесообразно будет добавить к этой библиотеке и файл с ошибками. Что позволит в одном месте хранить и описание ошибок и функции, которые могут их обрабатывать.

Чтобы получить динамическую библиотеку, необходимо создать пустой проект типа DLL библиотека, добавить уже скомпилированный файл (*.res). В настройках проекта(Настройки конфигурации->Компоновщик->Дополнительно) установить метку «без точки входа», и скомпилировать проект. После чего можно использовать полученную библиотеку(*.dll).

Разработка

На предыдущих этапах были получены файлы с описаниями собственных ошибок. Для примера реализации работы с ними предлагается рассмотреть небольшое приложение на языке Assembler, которое должно считывать данные из файла(имя файла передаётся в командной строке) и выполнять некоторую их обработку. Собственные ошибки будем хранить в динамической библиотеке, символьное описание кодов ошибок получим путём редактирования заголовочного файла и сохранения его в формате *.inc. Например, строку

```
#define EMPTY_CMD_LINE_ERROR
((DWORD)0xC0020001L)
```

заменяем на:

```
EMPTY_CMD_LINE_ERROR equ 0C0020001h
```

Описание переменных, подключение заголовочных файлов и основная процедура представлены на рисунке 3.

```
include masm32rt.inc
include my_err.inc
        .data
hModule dd 0
nameModule db "MYErrors.dll", 0
err_str dw 256 dup(?)
type_error db "Операция открытия файла.",0
CommandLine dd 0
argc dd 0
```

```

.code
start:
    invoke SetLastError, 0
    call CreateConsole
    invoke LoadLibraryA, offset nameModule
    mov hModule, eax ;подключение модуля с ошибками
    call convertCommandLine
    cmp argc, 1 ;количество аргументов в командной строке
    jg right_console
        invoke SetLastError, EMPTY_CMD_LINE_ERROR;установка ошибки
        lea edi, type_error
        call printError;проверка ошибки
        jmp end_w
    right_console:
        call work
    end_w:
        invoke FreeLibrary, hModule;освобождение файла с ошибками

```

Рисунок - 4. Данные программы и основная процедура

Процедура **SetLastError** загружает в стек ошибок код ошибки(dword значение). Подключение заголовочного файла позволило нам вместо кода ошибки писать его символьный эквивалент. Процедуры **LoadLibrary** и **FreeLibrary** осуществляют соответственно загрузку и освобождение модуля с ошибками из памяти.

Для проверки, произошла ли ошибка, и если да, то какая. Используется процедура, представленная на рисунке 4. Процедура **GetLastError** возвращает из стека ошибок в **eax** код последней погруженной ошибки и смещает указатель стека ошибок на поле одной ошибки. Для формирования кода существуют различные функции, например описанные в [4]. Но Рихтер в своей книге [2] рекомендует использовать процедуру **FormatMessage** как наиболее функциональную и удобную. Описание её аргументов содержится в таблице 5.

```

;edi -> type error[string]
printError    proc
    invoke GetLastError
    test eax, eax
    je no_error
        lea esi, err_str
        invoke FormatMessageA, FORMAT_MESSAGE_FROM_HMODULE + FORMAT_MESSAGE_FROM_SYSTEM,
            hModule, eax, 0, esi, 200, 0
        invoke MessageBox, 0, esi, edi, MB_OK + MB_ICONERROR
    no_error:
    ret
printError    endp

```

Рисунок - 5. Процедура для проверки ошибки

Таблица 5. Аргументы для процедуры **FormatMessage**

Параметр	Значение
FORMAT_MESSAGE_FROM_HMODULE + FORMAT_MESSAGE_FROM_SYSTEM	Флаги: формирование сообщения из системных сообщений или пользовательской библиотеки
hModule	Дескриптор модуля с описанием своих ошибок
eax	Код ошибки
0	Язык – язык системы(можно указывать код любого языка, который мы описывали в LanguageNames в *.mc файле)
esi	Запись сообщения в поле, адрес которого хранит esi
200	Максимальная длина поля сообщения
0	Форматированный вывод не используется

Демонстрация работы программы представлена на рисунке 5.

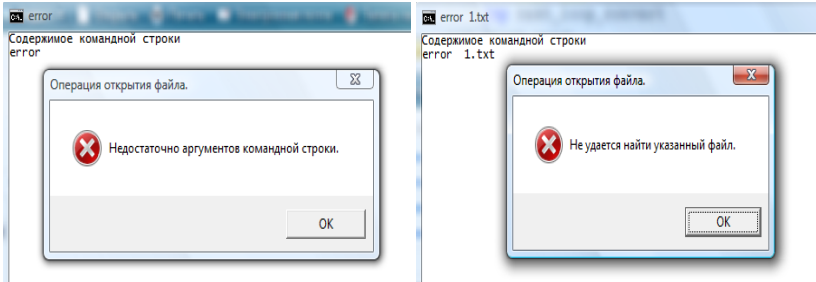


Рисунок - 6. Сообщение об ошибке (слева собственная ошибка, справа системная)

Выводы

В результате работы был рассмотрен вопрос обработки ошибок функций Windows. Также представлен метод описания собственных ошибок и использования его при написании приложений. Была разработана на языке Assembler программа, демонстрирующая обработку ошибок, описанных в winerror.h, а также промоделирована ситуация возникновения собственной ошибки. Рассмотрены варианты подключения ресурсного файла и динамической библиотеки. Проверка ошибок таким образом, как описано в статье позволяет производить одинаковую обработку как системных, так и пользовательских ошибок.

Список литературы

- [1] Финогенов К.Г. Win32. Основы программирования. – 2-е изд., испр. и дополн. – М.: ДИАЛОГ – МИФИ, 2006. – 416с.
- [2] Рихтер Дж. Windows для профессионалов: создание эффективных Win32 – приложений с учетом специфики 64 – разрядной версии Windows/ Пер. с англ. – 4-е изд. – Спб.: Питер; М.: Издательство «Русская Редакция»; 2008. – 720 стр.:ил.
- [3] Ресурс MSDN. / Интернет-ресурс. - Режим доступа : URL: <http://msdn.microsoft.com>
- [4] Список функций для получения текста ошибки. / Интернет-ресурс. - Режим доступа : URL: <http://habrahabr.ru/post/149116/>

УДК 004.4'236

Д.О. Макагонов, Н.А. Романовский, Д.М. Бочаров

Донецкий национальный технический университет,
кафедра программного обеспечения интеллектуальных систем

РАЗРАБОТКА И ПОДДЕЖКА ПРОГРАММНОГО МОДУЛЯ СИСТЕМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ И ЗАДАЧАМИ

Аннотация

Д.О. Макагонов, Н.А. Романовский, Д.М. Бочаров. Разработка и поддержка программного модуля системы управления проектами и задачами. Выполнен анализ систем управления проектами и задачами, а в частности серверное веб – приложение Redmine. Проведен анализ высокоуровневого языка программирования Ruby. Проанализированы аналоги и выведена целесообразность, и потребность создания системы.

Ключевые слова: клиент – серверное веб – приложение Redmine, язык программирования Ruby.

Министерство связи Донецкой Народной Республики - республиканский орган исполнительной власти.

Министерство связи руководствуется в своей деятельности Конституцией, законами, актами и указами Совета Министров, международными договорами, иными государственными нормативно-правовыми актами, а также временным Положением о Министерстве. Осуществляет свою деятельность во взаимодействии с другими органами исполнительной власти, органами местного управления, общественными объединениями и иными организациями Донецкой Народной Республики. Министерство связи выступает в качестве почтовой администрации, выполняет функции администрации связи при осуществлении международной деятельности в области связи, а также имеет административные права при осуществлении международной деятельности в сфере компьютерных телекоммуникаций.

Министерство связи Донецкой Народной Республики осуществляет координацию и контроль деятельности государственных предприятий, находящихся в его ведении, таких как:

- 1) Главное государственное архивное управление;
- 2) Министерство Здравоохранения;
- 3) Министерство Культуры;
- 4) Мобильный оператор Феникс;
- 5) Телефония ГП «Углетелеком».

Актуальность и необходимость разработки модуля назначения задач состоит в удобном, понятном и контролируемом процессом назначения новых проектов и задач, для обеспечения взаимодействия со всеми Министерствами и ведомствами внутри Донецкой Народной Республики.

При разработке и поддержке программного модуля, системы управления проектами, для Министерства Связи Донецкой Народной Республики, за основу было взято серверное веб - приложение Redmine, написанное на основе языка Ruby. Поскольку Redmine является открытым серверным веб - приложением, оно требует доработок, а так же внедрения новых функций, которые позволят расширить возможности приложения.

Системы планирования, постановки и контроля задач - основной инструмент работы над проектами в компаниях и предприятиях любого размера. В таблице 1 приведена сравнительная характеристика наиболее популярных таких систем.

Таблица 1 - Сравнение систем управления задачами

Оцениваемый параметр	Asana	Basecamp	JIRA	Redmine	Битрикс 24	Trello	Мегаплан
Бесплатный аккаунт	Есть	Нет	Нет	Есть	Есть	Есть	Есть
Дисковое пространство бесплатно	-	Нет	Нет	-	5 ГБ	-	300 МБ

Число подключенных пользователей бесплатно	От 1 до 14	Нет	Нет	-	12	-	От 1 до 7
Число проектов бесплатно	-	-	Нет	∞	1	∞	-
Период try & buy	Нет	60 дней	30 дней	Нет	Нет	Нет	14 дней
Платный аккаунт	Есть	Есть	Есть	Нет	Есть	Есть	Есть
Мин. стоимость платного пользования	\$50 / мес.	\$20 / мес.	\$10 / мес.	Пожертвования	\$50 / мес.	\$20 / мес.	\$10 / мес.
Макс. стоимость платного пользования	\$800 / мес.	\$150 / мес.	\$1500 / мес.		\$287 / мес.	\$45 чел./год	\$1824 / мес.

Redmine - бесплатный open - source инструмент для управления проектами и баг - трекинга с поддержкой диаграмм Ганта, визуализацией дедлайнов и прогресса по нескольким проектам одновременно. Написан на Ruby on Rails, работает как кроссплатформенное решение.

Благодаря тому, что Redmine является бесплатным кроссплатформенным инструментом управления проектами и задачами, с открытым исходным кодом, позволяющий следить за несколькими проектами одновременно, система была взята за основу управления в Министерстве Связи. Написанная на языке Ruby, который является динамическим, рефлексивным интерпретируемым высокоуровневым языком программирования, обеспечивающий быстрое и удобное объектно-ориентированное программирование. Независимость от операционной системы дает большое преимущество по сравнению с конкурентами. Данный язык программирования близок к Perl и Eiffel, а некоторые его части были взяты из таких языков как Python, Lisp, Dylan, что обеспечивает достаточно легкое понимание

синтаксиса программного кода. Кроссплатформенная реализация интерпретатора языка является полностью свободной.

Такой крупной государственной структуре как Министерство Связи для учёта и контроля задач требуется система управления проектами и задачами.

Основное меню управления проектами обеспечивает навигацию в приложении, позволяя перейти к назначенным задачам, создать новую или проверить совершаемые действия внутри отделов.

Административная часть меню назначения задач (см. рис.1) позволяет выбрать настраиваемые поля ввода для добавления новых пользователей в систему, обеспечивая при этом возможность назначения новых исполнителей.

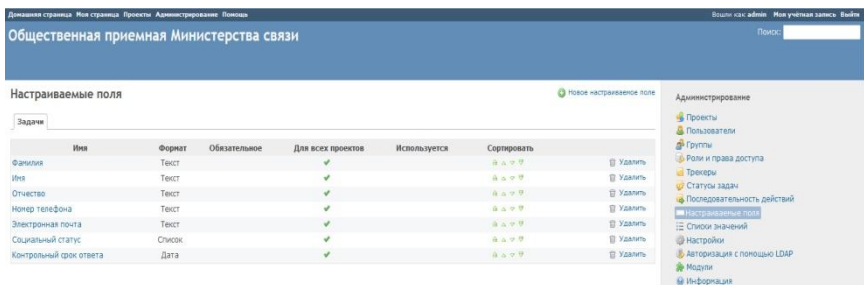


Рисунок 1 - Административная часть меню создания настраиваемых полей ввода

Административная часть меню навигации предоставляет доступ к настройке добавления новых проектов, новых пользователей, настройке присваивания прав и разрешения/запрета доступа к проектам. Так же позволяет настроить статус задачи и общий список значений.

Административная часть меню выбора созданных проектов обеспечивает навигацию по проектам внутри приложения, позволяя перейти к проекту и проверить статус его выполнения. Перейдя к выбранному проекту, можно просмотреть текущий статус выполнения, количество назначенных задач или же

просмотреть весь список. Справа отображаются назначенные на проект сотрудники и их руководители.

В меню просмотра назначенных задач меню можно просмотреть статус задачи, а так же увидеть её приоритет, который подсвечивается определенным цветом по важности.

Позволяет просмотреть список назначенных задач на проект, отсортировать задачи по теме, просмотреть автора задачи и увидеть назначенного исполнителя.

В поле фильтр предоставляется возможность отсортировать статусы задач и определить, на каком этапе на данный момент находятся задачи проекта, соответствуют ли они тематики проекта. Поле «закрыто» позволит выбрать закрытые задачи, если они не были удалены по их окончанию за ненадобностью.

Меню добавления новой задачи (см. рис.2) внутри выбранного проекта позволяет добавить новую задачу внутри выбранного проекта. Ввести тему задачи, описать задачу, поставить статус (изменяется на любом этапе выполнения задачи), и выбрать приоритет важности из выпадающего меню «Приоритет». Присутствует возможность связать задачи между собой, при помощи функции «Родительская задача». Назначить задачу на сотрудника из списка зарегистрированных в пункте «Назначена», указав при этом дату выполнения, в соответствующем меню «Дата выполнения». В дальнейшем сотрудник может указать в поле «Готовность», процент выполненной работы, и ввести в поле «Оценка времени» затраченное время на выполнение задачи. Эта функция позволяет в дальнейшем отображать на диаграммах и в календаре, сроки выполнения проектов, для дальнейшей оценки затраченного времени и позволит руководству оценить возможности для выполнения следующих проектов и задач.

Созданную задачу можно поместить в другой проект, при помощи функции «Отображается в проектах».

Для более удобного поиска задач в системе, присутствует возможность заполнения меток при создании задачи или же в процессе ее выполнения.

Большим плюсом системы, является возможность загрузки

Новая задача

Трекер * Задача

Тема *

Описание

Оставьте комментарий

Статус * Новая

Приоритет * Нормальный

Назначена

Родительская задача

Начата 2015-04-23

Дата выполнения

Оценка времени час(а,ов)

Готовность 0%

Отображается [Редактировать список проектов](#)

проектах *

Метки [+ добавить метку](#)

Файлы [Обзор...](#) Файлы не выбраны. (Максимальный размер: 500 МБ)

[Создать](#) [Создать и продолжить](#) [предпросмотр](#)

Рисунок 2 - Меню создания новой задачи внутри проекта.

файлов общим объемом, не превышающим 500 МБ.

Перед окончательным сохранением задачи, её можно предварительно просмотреть, при помощи функции «Предпросмотр».

Выводы. Результатом проделанной работы является система управления проектами и задачами, созданная с помощью серверного web-приложения Redmine. Рассматриваемая система служит для понятного и удобного распределения проектов внутри государственной структуры, а также обеспечивает комфортное слежение за прогрессом выполнения проектов.

Уполномоченные сотрудники, а так же руководители отделов, могут создавать и редактировать все проекты и задачи на стадии их выполнения.

Поддержка системы управления проектами и задачами для системы под названием «IT Incubator», позволит обеспечить

удобное взаимодействие всех пользователей, руководителей и участников проектов.

При работе с данными обеспечивается контроль целостности и сохранности данных, а также достоверности хранимой информации.

Список литературы

1. Д. Флэнаган, Ю. Мацумото. Язык программирования Ruby = The Ruby Programming Language / пер. с англ. Н. Вильчинский. — 1-е изд. — СПб.: Питер, 2011. — 496 с.

2. Статья о серверном веб - приложении Redmine, Свободная энциклопедия Википедия [Электронный ресурс]: Точка доступа URL

URL <https://ru.wikipedia.org/wiki/Redmine>

3. Образовательный ресурс, посвященный Ruby, Электронный ресурс]: Точка доступа URL <http://acm.mipt.ru/twiki/bin/view/Ruby>



**МОДЕЛИРОВАНИЕ И
СОВРЕМЕННЫЕ
КОМПЬЮТЕРНЫЕ
ТЕХНОЛОГИИ**

СЕКЦИЯ 7

УДК 517.988

А.С. Миненко

Донецкий национальный технический университет
Кафедра системного анализа и моделирования

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ОДНОГО КЛАССА НЕЛИНЕЙНЫХ СИСТЕМ С ПРИМЕНЕНИЕМ НЕЧЕТКОЙ ЛОГИКИ

Аннотация.

А.С. Миненко. Математическое моделирование одного класса нелинейных систем с применением нечеткой логики. Строится трехмерная математическая модель кристаллизации металла с учетом конвективного теплообмена. При управлении этим процессом используется нечеткая логика. Методом Рунца строятся приближенные решения, сходящиеся к точному решению в W_2^1 и C .

1. Рассмотрим область $\Omega = \{(x_1, x_2, x_3) : r^2 < x_1^2 + x_2^2 + x_3^2 < R^2\}$ и через Γ^- и Γ^+ обозначим следующие сферы:

$$\Gamma^- = \{(x_1, x_2, x_3) : x_1^2 + x_2^2 + x_3^2 = r^2\}, \Gamma^+ = \{(x_1, x_2, x_3) : x_1^2 + x_2^2 + x_3^2 = R^2\}.$$

Пусть Γ_0 – гладкая связная поверхность без самопересечений, лежащая внутри Ω , которая разбивает ее на две подобласти Ω^+ и Ω^- , т.е. $\Omega = \Omega^+ \cup \Omega^-$, причем сфера Γ^- лежит внутри ограниченной области, границей которой является Γ_0 . Рассмотрим краевую задачу со свободной границей Γ_0 . Требуется определить тройку $(u^\pm(x), \Gamma_0)$ по следующим условиям:

$$\begin{aligned} \nabla^2 u^\pm(x) &= 0, x \in \Omega^\pm; u^\pm(x) \Big|_{\Gamma^\pm} = B^\pm(x); \\ u^\pm(x) &= 1, |\nabla u^-(x)| - |\nabla u^+(x)| = 0, x \in \Gamma_0. \end{aligned} \quad (1)$$

При этом, $B^\pm(x) \in C^{3+\alpha}(\Gamma^\pm)$, $u^\pm(x) \in C^{3+\alpha}(\overline{\Omega^\pm})$, а Γ_0 принадлежит классу C^∞ [1].

Затем введем в рассмотрение функцию $u(x)$, заданную следующим образом $u = u^-(x)$ при $x \in \overline{\Omega^-}$ и $u = u^+(x)$ при $x \in \overline{\Omega^+}$. Тогда функцию $u(x)$ можно найти из условия минимума функционала $I(u, \Gamma_0) = \iint_{\Omega} |\nabla u|^2 dx_1 dx_2 dx_3$ на соответствующем множестве R допустимых функций [2]. Это следует из формулы первой вариации интегрального функционала с неизвестной областью интегрирования [2].

Далее, удобно представить функционал I в сферических координатах:

$$I(u, \Gamma_0) = \int_0^{2\pi} \int_0^\pi \int_r^R \left(u_\rho^2 + \frac{1}{\rho^2} u_\theta^2 + \frac{1}{\rho^2 \sin^2 \theta} u_\varphi^2 \right) \rho^2 \sin \theta d\varphi d\theta d\rho. \quad (2)$$

Лемма 1. Пусть тройка $(u^\pm(x), \Gamma_0)$ является классическим решением задачи (1). Тогда эта тройка будет стационарной для функционала (2) на множестве R . Обратно, каждая стационарная тройка $(u^\pm(x), \Gamma_0)$ функционала (2) на множестве R , где Γ_0 – достаточно гладкая связная поверхность, является решением задачи (1).

Сформулированная задача (1) получается из задачи, изученной в [1] в случае $\vec{V} = 0$, т.е. в случае бесконечно большой вязкости, $Re = 0$. Поэтому в дальнейшем под решением задачи (1) при $Re = 0$ будем понимать функции $\vec{V}(x) = 0, u^+(x)$ и $u^-(x)$, заданные в Ω^\pm . Из условий (1) следует, что Γ_0 – не что иное, как линия уровня функции $u(x)$, т.е.:

$$\Gamma_0 = \{x \in \Omega : u(x) = 1\}.$$

Если предположить выполнение условия:

$$\pm(B^\pm(x) - 1) \geq \varepsilon_0 > 0, x \in \Gamma^\pm,$$

где ε_0 – некоторая постоянная, тогда поверхность Γ_0 лежит внутри области Ω и представляет собой поверхность класса $C^{4+\alpha}$, не имеющую самопересечений и располагающую относительно Γ^+ и Γ^- аналогично поверхности Γ_t (свободная поверхность), изученной в [1]. Следовательно, рассматривая функцию $u(x)$ в одной из областей Ω^\pm и принимая во внимание лемму о нормальной производной, находим, что: $\frac{\partial u}{\partial n} = |\nabla u| \geq \varepsilon > 0, x \in \Gamma_0$,

где n – нормаль к Γ_0 , направленная в сторону Ω_0^+ , а ε – некоторая постоянная. Отсюда, применяя теорему о неявной функции, следует, что Γ_0 принадлежит классу C^∞ , так как этому классу в некоторой окрестности Γ_0 принадлежит гармоническая функция $u(x)$.

2. Минимум функционала (2) на множестве R будем искать при помощи сумм:

$$u_n = B^+ + \frac{R^2 - \rho^2}{R^2 - r^2} (B^- - B^+) + (R^2 - \rho^2)(\rho^2 - r^2) \sum_{k=0}^n C_k \rho^k y_k(\varphi, \theta),$$

где $y_k(\varphi, \theta)$ – сферические функции. Неизвестные коэффициенты C_k определяют при помощи метода Ритца. Тогда поверхность $\Gamma_0 : \rho = \rho_0(\varphi, \theta)$ определяется из уравнения $u_n(\varphi, \theta, \rho_0(\varphi, \theta)) = 1$. При этом необходимо учесть, что $|\nabla u(x)| \geq \varepsilon_0 > 0$ в $\bar{\Omega}$, где ε_0 – некоторая постоянная [2].

Лемма 2. При малых t справедливо представление

$$\Gamma_t : \rho(\varphi, \theta, t) = \rho_0(\varphi, \theta) - \operatorname{Re} \frac{u_1^\pm(\varphi, \theta, t)}{|\nabla A^\pm(\varphi, \theta)|} + o(\operatorname{Re}), (\varphi, \theta) \in \Gamma_0. \quad (3)$$

Здесь Re – число Рейнольдса, а $u_1^\pm(\varphi, \theta, t)$ – первое приближение исходной задачи, изученной в [1].

В частности, для нулевого приближения $u_0(\varphi, \theta)$ из уравнения $u_0 = B^+ + \frac{R^2 - \rho^2}{R^2 - r^2}(B^- - B^+) + (\rho^2 - r^2)(R^2 - \rho^2)C_0 = 1$ легко найти поверхность $\rho_0(\varphi, \theta)$.

Рассмотрим величину $\varepsilon_n = I(u_n, \Gamma_0) - I(u, \Gamma_0)$, где u – точное решение задачи (1). Тогда можно установить, что $\varepsilon_n \rightarrow 0$ при $n \rightarrow \infty$, если C_k – коэффициенты Рунца. Используя затем результаты Канторовича Л.В. по минимизации квадратичных функционалов аналогично тому, как это сделано в [1], можно доказать следующее утверждение.

Теорема. *Последовательность приближений Рунца u_n сходится к решению задачи (1) и по норме в W_2^1 и C , причем*

$\varepsilon_n = O\left(\omega^{(3)}\left(u, \frac{1}{n}\right) / n^2\right)$, и если $\lim_{n \rightarrow \infty} \omega^{(3)}\left(u, \frac{1}{n}\right) n^{-1} (\ln n)^{1+\varepsilon} = 0$, то

$$\|u - u_n\|_{C(\bar{\Omega})} \leq C_1 \omega^{(3)}\left(u, \frac{1}{n}\right) \cdot n^{-1} + C_2 \sum_{s=m}^{\infty} \omega^{(3)}\left(u, \frac{1}{2^s}\right) \cdot 2^{-s},$$

где C_1 и C_2 – некоторые постоянные; $\omega^{(3)}\left(u, \frac{1}{n}\right)$ – максимальный модуль непрерывности производных третьего порядка функции $u(x)$ и $2^{m-1} \leq m < 2^m$.

Замечание. В случае двух геометрических переменных имеют место оценки:

$$\varepsilon_n = O\left(\frac{1}{n^{2(2+\alpha)}}\right), \|u_n - u\|_{C(\bar{\Omega})} \leq C_1 \sqrt{\varepsilon_n \ln \frac{n}{\varepsilon_n}} + C_2 \sqrt{\varepsilon_n}, \quad (4)$$

В работе [1] изучены k -е приближения $(\vec{V}, u_k^\pm, \rho_k)$ исходной задачи, являющиеся функциями класса $H^{2+\alpha, (2+\alpha)/2}(\bar{\Omega}^\pm)$, построены системы уравнений, решениями которых они являются. Формулы (3), (4) позволяют исследовать Γ , в зависимости от Re .

3. Пусть T^* – температура, которую должна достичь поверхность $\partial\Omega$. Эта температура достигается за счет воздействия тепловых потоков мощности w_1, w_2, w_3 , причем мощность одного из них w_3 равномерно распределена в центре $\partial\Omega$, а два других w_1 и w_2 сконцентрированы по краям $\partial\Omega$ [3]. Далее будет предложен метод нечеткого управления в данном классе задач, который имеет место в спецметаллургии [4].

Пусть X_1, X_2, \dots, X_n – факторы, влияющие на процессе кристаллизации, а Y_1, Y_2, \dots, Y_n – условия, при которых происходит появление нового слитка. Тогда нечеткое управление в нашей модели можно представить в виде функционального отображения $X = \{X_1, X_2, \dots, X_n\} \rightarrow Y = \{Y_1, Y_2, \dots, Y_n\}$.

Поверхность $\Gamma_0 : \rho = \rho_0(\varphi, \theta)$ можно найти из условия $u_0(\varphi, \theta, \rho_0(\varphi, \theta)) = 0$. Тогда для поверхности Γ_t можно воспользоваться уравнением [2]:

$$\Gamma_t = \rho(\varphi, \theta, t) = \rho_0(\varphi, \theta) - \operatorname{Re} \frac{u_1^+(\varphi, \theta, t)}{|\nabla A^+(\varphi, \theta)|} + o(\operatorname{Re}).$$

Численная реализация задачи (1) была проведена при следующих значениях параметров: $t = 200$, $R = 6$, $r = 0,8$, $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{3}$, $-\frac{\pi}{2} \leq \varphi \leq \frac{\pi}{2}$, $B^+ = 3[\cos^2 \theta + \cos^2 \varphi]$, $B = -0,35[\cos^2 \theta + \cos^2 \varphi] - 0,1$. Свободная поверхность Γ_t расположена между сферами радиусов R и r .

При численной реализации нечеткого управления использовались следующие значения параметров:

$$2500 \text{ MBm} / \text{m}^2 \leq W \leq 5000 \text{ MBm} / \text{m}^2, 600 \text{ мм} \leq L \leq 6000 \text{ мм}.$$

Численный расчет, позволяющий построить нечеткое управление, был осуществлен с помощью стандартного алгоритма Мамдани, а результаты получены в ходе эксперимента на объектах управления ЭСП [3].

Выводы. Построена новая математическая модель нелинейной проблемы, описывающая кристаллизацию металла. Предложен новый алгоритм построения поверхности кристаллизации. Предлагается вариант управления этим процессом при завершении выращивания слитка при кристаллизации основанный на нечеткой логике.

Список литературы

1. Шевченко А.И., Миненко А.С. Задача Стефана при наличии конвекции//Доп. НАН України. – 2012. – №1. – С.25-29.
2. Миненко А.С. Вариационные задачи со свободной границей. – Киев: Наук.думка, 2005. – 341 с.
3. Патон Б.Е. Избранные труды. – Киев: Институт электросварки им. Е.О. Патона НАН Украины, 2008. – 893 с.
4. Шевченко А.И., Миненко А.С. Методы исследования нелинейных математических моделей. – Киев: Национальная академия наук Украины, ИПИИ, 2012 – 130 с.

УДК 004

Подлесный А.И., Григорьев А.В.

Донецкий национальный технический университет

Кафедра прикладной математики и информатики

E-mail: podliy16@gmail.com, grigorievalv1@gmail.com

ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА В СИСТЕМАХ АВТОМАТИЧЕСКОГО ПРОЕКТИРОВАНИЯ

Аннотация

Подлесный А.И., Григорьев А.В. Применение генетического алгоритма в системах автоматического проектирования. В данной статье описывается возможность применения генетического алгоритма в области проектирования. Приведен пример использования данной методики для проектирования ходовой части подводной лодки. Проведен анализ с другими методами оптимизации.

Актуальность

Автоматизация производственных процессов – важный двигатель технического прогресса. Системы автоматического проектирования пользуются все большей популярностью у инженеров и значительно сокращают затраты времени для проектирования. Идея генетического алгоритма «подсмотрена» у природы и использует механизмы аналогичные естественному отбору.

Изложение материалов исследования

Многие реальные задачи можно свести к задаче нахождения экстремума целевой функции, где целевая функция – сложная функция, зависящая от некоторых входных параметров. Задачи проектирования также можно свести к оптимизации такой

функции. В качестве примера я покажу проектирование ходовой части подводной лодки в среде разработки Matlab.

Для применения данного метода нужно определить функцию, которую мы будем оптимизировать, и изначальную популяцию, которая состоит из уже реализованных проектов. В качестве функции я буду рассматривать максимальную скорость лодки, которая зависит от размера винта, скорости вращения двигателя, мощности двигателя, поступи винта и массы лодки. Начальная популяция – собранные данные по уже готовым лодкам.

Затем мы применим генетический алгоритм для поиска оптимальных параметров заданной функции. В общем случае генетический алгоритм – оптимизация полного перебора, но при этом в алгоритме используются сильнейшие механизмы – мутация и селекция, которые позволяют осуществлять направленный поиск.

В результате работы всего алгоритма мы получим вектор входных параметров дающих оптимальное или субоптимальное решение исходной задачи. В зависимости от конфигурации алгоритма решения могут состоять ТОЛЬКО из уже существующих механизмов, например в результате работы мы получим новую подводную лодку, которая работает на двигателе лодки №3, с винтом лодки №24 и форму лодки № 56. Либо же мы на выходе мы получим характеристики, которыми должны обладать механизмы для достижения максимальной скорости хода.

Выводы

По итогам данной статьи можно увидеть, что генетический алгоритм является пригодным средством для автоматического проектирования. Если построить полную модель моделируемого средства, которая будет учитывать все тонкости и нюансы, то мы сможем создавать новые виды продукции, указывая, лишь требования к новому продукту.

Основные преимущества генетического алгоритма, по сравнению с другими методами:

-- «прямой» метод работы (алгоритм требует только вычисление функции в точках приближения). Прямой метод работы позволяет рассчитывать сложные модели, которые невозможно описать аналитически.

-- контролируемая изначальная популяция, с помощью которой мы в какой-то степени обучаем алгоритм на выборке уже реализованных проектов.

На данный момент такая возможность кажется слишком оптимистичной, но я считаю, что в ближайшем будущем будут появляться инструменты, предоставляющие схожую функциональность.

Литература

1. Емельянов В.В., Курейчик В.В., Курейчик В.М. Теория и практика эволюционного моделирования. – М: Физматлит, 2003. – С. 432.
2. Генетический алгоритм. Просто о сложном: [электронный ресурс]. Режим доступа: <http://habrahabr.ru/post/128704/>
3. Genetic Algorithm: [электронный ресурс]. Режим доступа: <http://mathworks.com/discovery/genetic-algorithm.html>

УДК 512.07

Ю. О. Сандул

Донецкий национальный технический университет, г. Донецк
кафедра системного анализа и моделирования

МОДЕЛЬ ИСПОЛНИТЕЛЕЙ СЛУЖБЫ ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ И РЕМОНТА ОБОРУДОВАНИЯ

Аннотация

Сандул Ю. О. Модель исполнителей службы технического обслуживания и ремонта оборудования. Ключевые слова: техническое обслуживание и ремонт, система управления техническим состоянием, система поддержки принятия решений, модель исполнителя.

Постановка проблемы. Целевое назначение службы ТОР можно фактически сформулировать так [1]: поддержание множества параметров, характеризующих техническое состояние оборудования, на уровне, предусмотренном нормативно-технической документацией, в течение всего периода использования оборудования по назначению за счет целенаправленных действий человека (группы людей) и орудий труда. В такой формулировке процесс технического обслуживания и ремонта полностью соответствует понятию «управление».

Следовательно, службу ТОР можно рассматривать как систему управления техническим состоянием оборудования (СУТС). Известно, что причиной отказа оборудования при его использовании по назначению является несвоевременное и некачественное проведение работ по техническому обслуживанию и ремонту. Следовательно, с точки зрения повышения эффективности работ по ТОР актуальным является управление техническим состоянием объекта в режиме «Профилактика».

Служба ТОР является организационной системой, на всех иерархических уровнях которой, по определению [2], находятся активные элементы (люди). Понятно, что и руководство (центр), и исполнители, как элементы СУТС, имеют единое представление о целевом назначении службы ТОР, однако их представления о способах достижения глобальной цели в том или ином конкретном случае могут существенно отличаться в силу различия их намерений, предпочтений, мотиваций и т.п.

Столкновение указанных представлений приводит к возникновению конфликта (конфликтной ситуации) [69], который может носить и скрытый характер. Известно, что упрощенной математической моделью конфликтной ситуации является игра [3]. Следовательно, решение поставленной задачи необходимо искать как решение игры, моделирующей взаимодействие центра и исполнителей в процессе их совместного функционирования. Для этого, в свою очередь, необходимо разработать и исследовать ряд моделей, одна из которых модель исполнителей.

Цель статьи - повышение эффективности управления профилактическим обслуживанием объектов регулярно-периодического использования за счет исследования и разработки системы поддержки принятия решений, включающей в себя модели формирования плана проведения и организации выполнения профилактик в различных условиях.

Модель исполнителя. Предварительно сделаем следующие предположения.

1. Полагаем, что каждый исполнитель обладает полными знаниями о технологии работы объекта (в бездефектном и дефектном состояниях) и его схемном или конструктивном исполнении.

2. Введено понятие обстановки $x \in X$, под которой понимается [2] состояние окружающей среды и поведение других исполнителей, работающих совместно с данным исполнителем. При этом окружающая среда – это пространство,

непосредственно примыкающее к объекту, а параметрами ее состояния могут быть, например, температурный режим, стесненность рабочего пространства, наличие работающего оборудования, находящегося в непосредственной близости от объекта, и т.д. Поскольку здесь рассматривается модель одного исполнителя, то в качестве обстановки примем только состояние окружающей среды.

3. Считаем, что по мере накопления опыта у исполнителя вырабатываются «свои» процедуры восстановления работоспособности, то есть некоторые стереотипы (в хорошем понимании этого слова). Причем, каждый такой стереотип можно рассматривать как индивидуальную норму в том смысле, что исполнитель сам сформировал действия, которые он считает необходимым реализовать в той или иной ситуации. Важно при этом отметить, что индивидуальная норма не выходит за рамки требований, установленных на конкретном предприятии и признанных всеми (например, требований безопасности). С учетом сказанного полагаем, что у каждого исполнителя сформировалась «своя» последовательность операций по обслуживанию k -го элемента объекта, то есть процедуру A_{kn} можно представить в виде:

$$A_{kn} = \langle a_{kn_1}, a_{kn_2}, \dots, a_{kn_{j,\dots}} \rangle \quad (1)$$

При этом заметим, что для каких-то фиксированных ситуаций порядок следования операций также строго фиксирован, что позволяет рассматривать процедуру (1) как кортеж. В общем же случае последовательность операций при выполнении процедуры A_{kn} может меняться в зависимости от конкретной обстановки, учитывая присущее человеку свойство адаптации.

4. Для определения времени t_k° выделим на оси времени t интервал $(t_{k_1}^\circ, t_{k_2}^\circ)$ границами которого являются минимальное время выполнения процедуры A_{kn} , обусловленное физическими возможностями исполнителя $t_{k_1}^\circ$, и время выполнения указанной

процедуры, установленное нормативными документами на данный объект $t_{k_2}^{\circ}$ [4].

Вначале рассмотрим детерминированный случай. Полагаем, что n -му исполнителю известна конкретная обстановка $x^- \in X$ при которой он должен будет восстановить работоспособность k -го элемента объекта.

Получив задание, n -й исполнитель, как активный элемент системы, планирует свое будущее поведение при восстановлении работоспособности k -го элемента объекта. Для этого он мысленно «проигрывает» процедуру A_{kn} , занимая при этом по отношению к этому «проигрыванию» позицию стороннего наблюдателя. Этот процесс называется авторефлексией [4]. При этом исполнитель не только формирует программу своего поведения, но и проводит самооценку, то есть устанавливает для себя время t_{kn}° , которое он предполагает затратить на выполнение процедуры A_{kn} . При этом рассуждения исполнителя можно представить так [4]: «Я уже выполнял процедуру A_{kn} в подобных условиях и знаю, что мне понадобится время, не меньшее, чем $t_{k_1}^{\circ}$, но и не большее, чем $t_{k_2}^{\circ}$; эту процедуру я реализую за промежуток времени, примерно, от $\tau_{k_1}^{\circ}$ до $\tau_{k_2}^{\circ}$ ».

Поскольку интервал $(t_{k_1}^{\circ}, t_{k_2}^{\circ})$ устанавливается самим n -м исполнителем, то можно утверждать, что этот интервал полностью соответствует его возможностям и предпочтениям. Следовательно, указанный интервал можно рассматривать в качестве нормы n -го исполнителя [4]. При этом функция принадлежности переменной t_{kn}° к понятию (терму) «Норма» будет равна:

$$\mu_N(\tau_{kn}^{\circ}) = \begin{cases} 0, \tau_{kn}^{\circ} \leq t_{k_1}^{\circ}; \\ \frac{\tau_{kn}^{\circ} - t_{k_1}^{\circ}}{\tau_{kn_1}^{\circ} - t_{k_1}^{\circ}}, t_{k_1}^{\circ} < \tau_{kn}^{\circ} < \tau_{kn_1}^{\circ}; \\ 1, \tau_{kn_1}^{\circ} \leq \tau_{kn}^{\circ} \leq t_{k_2}^{\circ}; \\ \frac{t_{k_2}^{\circ} - \tau_{kn}^{\circ}}{t_{k_2}^{\circ} - \tau_{kn_2}^{\circ}}, \tau_{kn_2}^{\circ} < \tau_{kn}^{\circ} < t_{k_2}^{\circ}; \\ 0, \tau_{kn}^{\circ} \geq t_{k_2}^{\circ}. \end{cases}$$

(2)

Таким образом, n -й исполнитель позиционирует себя как специалист, нормой поведения которого при обслуживании k -го элемента объекта в условиях определенности является выполнение процедуры A_{kn} за промежуток времени от $\tau_{kn_1}^{\circ}$ до $\tau_{kn_2}^{\circ}$.

Теперь предположим, что n -му исполнителю неизвестна обстановка, при которой он должен будет восстановить работоспособность k -го элемента объекта. Но, поскольку рассматривается конкретный объект обслуживания, то с полной уверенностью можно утверждать, что, как минимум, граничные «значения» обстановки являются априори известными. Более того, можно утверждать, что $x_{min} \in X$ (наиболее благоприятные условия) и $x_{max} \in X$ (наименее благоприятные условия) являются постоянными величинами. Тем самым имеем классическую задачу принятия решений в условиях интервальной неопределенности.

Известно, что единственным объективным решением данного класса задач является так называемый гарантированный результат, то есть результат, который может быть получен при наихудших условиях.

Будем полагать, что n -й исполнитель также придерживается указанного принципа. Далее положим, что временной интервал

$(t_{k_1}^{\circ}, t_{k_2}^{\circ})$ охватывает все возможные ситуации, связанные с обстановкой. И, наконец, учтем, что при определении оценок τ_{kn}° исполнитель рассуждал так: «Я уже выполнял процедуру A_{kn} в подобных условиях ...». Тем самым можно предположить, что интервал $(\tau_{kn_1}^{\circ}, \tau_{kn_2}^{\circ})$ является постоянной величиной, но смещается в пределах интервала $(t_{k_1}^{\circ}, t_{k_2}^{\circ})$ в зависимости от обстановки.

Если с учетом всего сказанного положить, что наилучшим условиям соответствует некий интервал оценок $(\tau_{kn_3}^{\circ}, t_{k_2}^{\circ}) \in (t_{k_1}^{\circ}, t_{k_2}^{\circ})$, то нормой n -го исполнителя (гарантированным результатом) в условиях неопределенности является время

$$\tau_{kn}^{\circ} \in (\tau_{kn_1}^{\circ}, \tau_{kn_2}^{\circ}) \in (\tau_{kn_3}^{\circ}, t_{k_2}^{\circ}) \mid \mu_N(\tau_{kn}^{\circ}) = 1. \quad (3)$$

Следовательно, в условиях неопределенности n -й исполнитель позиционирует себя как специалист, нормой поведения которого при техническом обслуживании k -го элемента объекта является выполнение процедуры A_{kn} за промежуток времени от $\tau_{kn_1}^{\circ}$ до $\tau_{kn_2}^{\circ}$, наиболее близко расположенный к правой границе интервала $(t_{k_1}^{\circ}, t_{k_2}^{\circ})$.

Данные модели сформированы без учета типа исполнителя, то есть без учета его уровня самооценки. В то же время вполне обоснованно можно предположить, что n -й исполнитель характеризуется завышенной, заниженной или адекватной самооценкой.

Понятно, что в случае завышенной самооценки интервал самооценок $(\tau_{kn_1}^{\circ}, \tau_{kn_2}^{\circ})$ будет смещен в сторону нижней границы интервала $(t_{k_1}^{\circ}, t_{k_2}^{\circ})$, а при заниженной самооценке – в сторону верхней границы интервала $(t_{k_1}^{\circ}, t_{k_2}^{\circ})$. Представим это в таблице 3.2.

Таблица 3.2 - Модель БВУ

		Исполнители				
		1	...	n	...	N
Элементы	1	$(\tau_{111}^{\circ}, \tau_{112}^{\circ})$...	$(\tau_{1n1}^{\circ}, \tau_{1n2}^{\circ})$...	$(\tau_{1N1}^{\circ}, \tau_{1N2}^{\circ})$

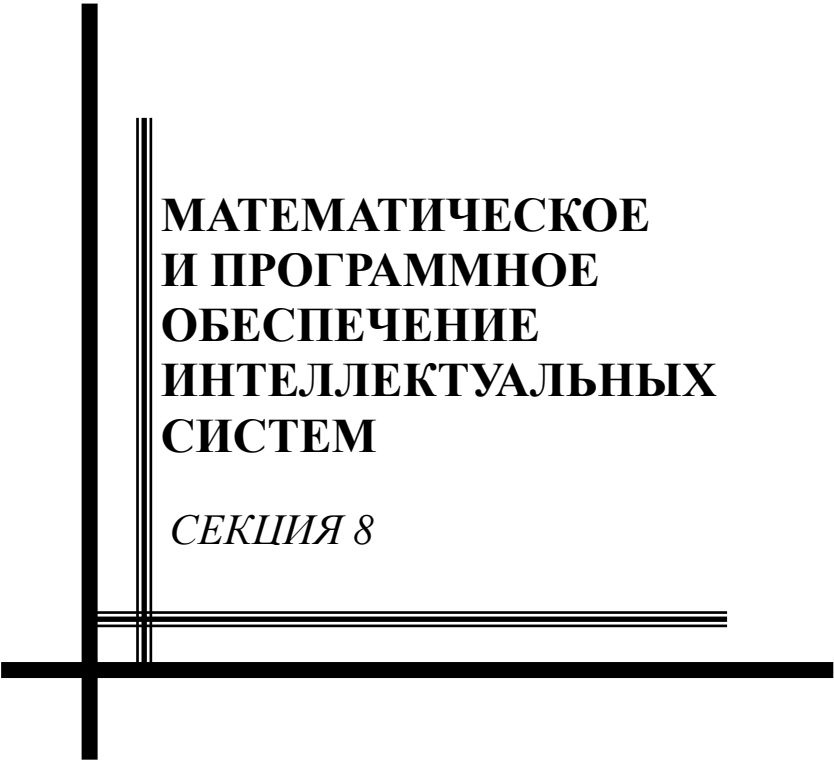
	k	$(\tau_{k11}^{\circ}, \tau_{k12}^{\circ})$...	$(\tau_{kn1}^{\circ}, \tau_{kn2}^{\circ})$...	$(\tau_{kN1}^{\circ}, \tau_{kN2}^{\circ})$

	K	$(\tau_{K11}^{\circ}, \tau_{K12}^{\circ})$...	$(\tau_{Kn1}^{\circ}, \tau_{Kn2}^{\circ})$...	$(\tau_{KN1}^{\circ}, \tau_{KN2}^{\circ})$

Выводы. Разработка и исследование модели исполнителей позволит избежать конфликтных ситуаций между руководством (центром) и исполнителями СУТС в силу различия своих намерений или мотиваций, не смотря на то, что они имеют единое представление о целевом назначении службы TOP.

Список литературы

1. Резников В.А., Суворова А.М. Качественные модели системы управления техническим состоянием оборудования // Искусственный интеллект. – 2011. – № 1. – С. 229-235.
2. Бурков В.Н., Коргин Н.А., Новиков Д.А. Введение в теорию управления организационными системами. – М.: Либроком, 2009. – 264 с.
3. Садовин Н.С., Садовина Т.Н. Основы теории игр: Учебное пособие. □ Йошкар-Ола: Изд-во МарГУ, 2011. – 119 с.
4. Резников В.А., Темник А.М. Нечёткая модель исполнителя системы технического обслуживания и ремонта в условиях определенности // Радіоелектронні і комп'ютерні системи. – 2012. – № 4. – С. 201-205.



**МАТЕМАТИЧЕСКОЕ
И ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ
ИНТЕЛЛЕКТУАЛЬНЫХ
СИСТЕМ**

СЕКЦИЯ 8

УДК 004.415

И.С. Гавага, И.А. Тарасова

Донецкий национальный технический университет

Факультет компьютерных наук и технологий

Кафедра программного обеспечения интеллектуальных систем

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ УПРАВЛЕНИЯ ПАТРУЛЬНО-ПОСТОВОЙ СЛУЖБОЙ МИЛИЦИИ

Аннотация

Гавага И.С., Тарасова И.А. Разработка программного обеспечения системы управления патрульно-постовой службой милиции. В статье обоснована актуальность и необходимость разработки компьютерной системы управления силами патрульно-постовой службы милиции, проанализированы существующие системы, используемые для управления патрульно-постовой службой милиции, разработано программное обеспечение системы управления патрульно-постовой службой милиции.

Ключевые слова: программное обеспечение, управление патрульно-постовой службой, геоинформационная система, база данных, анализ данных.

Общая постановка задачи.

Патрульно-постовая служба милиции – это деятельность специальных нарядов милиции по охране гражданского порядка, безопасности и борьбы с правонарушениями на улицах, площадях, в парках, на транспортных магистралях, портах, в аэропортах и других гражданских объектах, а также при проведении массовых мероприятий, ликвидации последствий аварий, катастроф, стихийных бедствий. Возрастающие объемы автотранспорта, качество и доступность вооружения, изменения криминогенной обстановки предъявляют высокие требования к техническому обеспечению отделов патрульно-постовых служб милиции. Важную роль в решении этих задач играет создание

автоматизированных систем управления. Разработка таких систем остается актуальной на протяжении многих лет. Это объясняется непрерывными изменениями оперативной обстановки, несмотря на достижения в возможностях технического обеспечения патрульно-постовых служб, а также новыми возможностями аппаратного и программного обеспечения, задействованного в подобной сфере деятельности. К настоящему времени разработано некоторое количество систем управления силами патрульно-постовых служб. К ним можно отнести такие системы, как СУМН-М (система управления мобильными нарядами милиции), центр геоинформационных систем для обеспечения безопасности, SityGIS (городской геоинформационный центр) и некоторые другие. Основой подобных систем является использование геоинформационных систем. Анализ существующих систем показал, что одним из основных их достоинств является автоматизация процесса обработки информации с последующим решением таких задач, как планирование патрульно-постовой службы милиции, управление задействованными в патрульно-постовой службе нарядами милиции, отображение на карте оперативной обстановки на основании введенных ранее данных. Ранее, при составлении плана патрульно-постовой службы, на этапе сбора данных происходил сбор всевозможной информации, связанной с происшествиями в городе. Данный процесс происходил с помощью подручных средств отдела письменном, распечатанном, электронном и тому подобном виде. Помимо этого, к достоинствам относятся отображение маршрутов и параметров подконтрольных объектов на электронной карте, создание маршрутов, одновременное наблюдение за десятками объектов с возможностью автоматического слежения, ведение баз данных маршрутов и объектов.

Тем не менее, несмотря на перечисленные выше достоинства и возможности подобных программно-технологических комплексов и программного обеспечения они не лишены и недостатков. Большая часть подобных программ способна серьезно выполнять лишь небольшое количество функций, что

оптимизирует и улучшает процесс работы службы милиции не на столь высокий уровень. Стоит также отметить и высокие требования к аппаратной части – лишь небольшая часть подобного программного обеспечения способна работать на персональном компьютере, без необходимости подключать аппаратное обеспечение сервера, которое и будет отвечать за полноценную работоспособность программно-технологического комплекса. Подобное ограничение резко повышает требования к использованию – покупка необходимых устройств, обучение персонала новым технологиям. К актуальным недостаткам стоит отнести большую зависимость от исходных географических данных, их точности и четкости их переноса в ГИС. Данные проблемы были выявлены при анализе доступных на рынке вариантов подобных программно-технологических комплексов и программного обеспечения.

Целью является повышение эффективности работы патрульно-постовой службы милиции за счет разработки программного обеспечения системы управления патрульно-постовой службой милиции.

Разработка программного обеспечения.

Программное обеспечение системы управления силами патрульно-постовой службы милиции разработано с целью повышения продуктивности работы отдела гражданской защиты. С целью достижения эффективного управления силами патрульно-постовой службы милиции создана компьютерная система, с помощью которой на основе поступающих входных данных от оператора определяются оптимальные маршруты патрулирования, точки дежурства и их продолжительность. Система реализует следующие основные функции:

- графический интерфейс пользователя;
- работа с базой данных программного обеспечения;
- отображения актуальной карты на основе информации из базы данных программного обеспечения;
- создание плана маршрута для патрульно-постовой службы.

Каждая из перечисленных функций выполняет определенные действия.



Рисунок 1 – Функциональная схема системы управления

Графический интерфейс пользователя отображает карту города, на которой отмечены маршруты патрулирования, точки дежурства и их продолжительность. Он выполнен в интуитивно-понятном стиле. Данное решение позволит избежать необходимости долгого обучения персонала и необходимости составлять подробное техническое руководство пользователя. Пользователь может в любой момент отредактировать маршрут, внося изменения по мере необходимости и обновления оперативной обстановки. В окне графического интерфейса отображается:

- карта;
- актуальные маршруты патрулирования;
- последние записи и обновления базы данных;

База данных системы управления выполняет функцию хранения и организации информации, необходимой для корректной работы программного обеспечения. Возможность редактировать и вносить новые записи позволяет сохранять актуальность карты, отображающей оперативную обстановку, и

создаваемых системой оптимальных маршрутов патрулирования, точек дежурств и их продолжительности. База данных системы управления содержит записи о преступлениях и нарушениях, которые заносит оператор. Данные можно в любой момент просмотреть. На основе занесенной оператором информации программное обеспечение создает оптимальные маршруты для патрульно-постовой службы милиции.

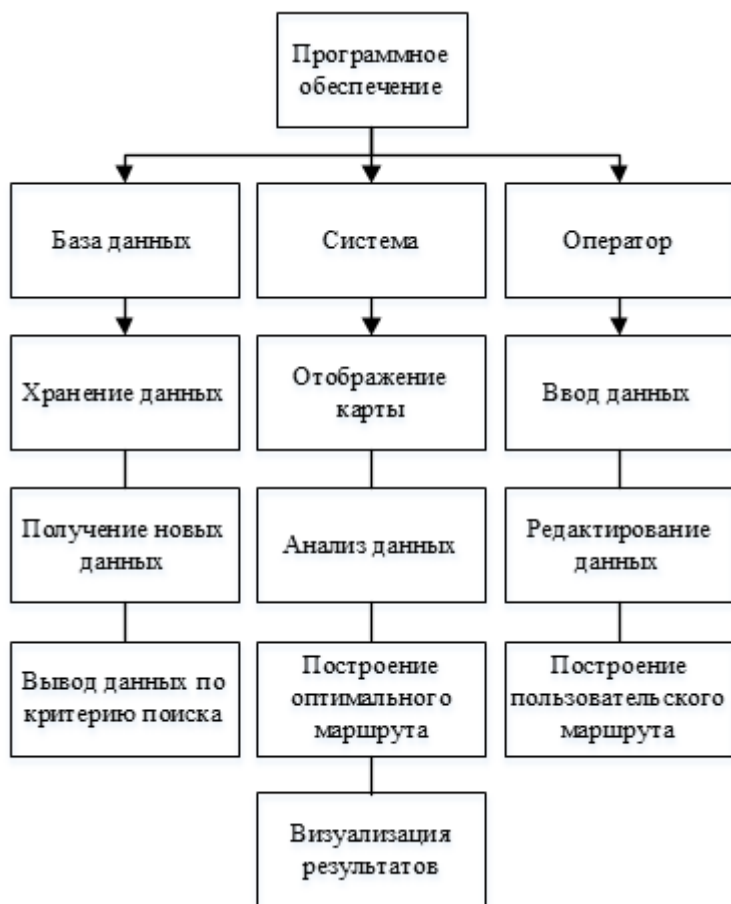


Рисунок 2 – Функциональная структура системы управления

Исходя из устава патрульно-постовой службы милиции Украины, для успешного выполнения возложенных на патрульно-постовую службу задач должна обеспечиваться рациональная расстановка нарядов; для этого потребуется проделать большой объем работ с бумажными данными и вручную определять расстановки. Данный процесс может занять продолжительное время, а результат на момент выполнения утратить свою актуальность в связи с изменением оперативной обстановки в округе. Разработанная система обрабатывает данные автоматически и обеспечит оптимальную расстановку нарядов милиции с учетом возможности своевременного маневра при поступлении вызова в милицию. В случае необходимости или изменения оперативной обстановки пользователь может изменить предложенный маршрут, внося поправки либо полностью создать свой вариант. Предложенный оптимальный маршрут отображается на карте в графическом интерфейсе пользователя. Количество маршрутов может превышать рассчитанное по обязательным нормам выставления нарядов. В таких случаях они делятся на подлежащие обязательному закрытию и такие, которые закрываются в зависимости от оперативной обстановки, которая складывается в течении определенного времени (от одного до нескольких дней).

Одним из основных достоинств разработанного программного обеспечения системы управления силами патрульно-постовой службы являются низкие системные требования к аппаратной части обеспечения. Это повышает доступность и практичность использования данного продукта. Одной из основных целей разработки было минимизировать нагрузку при работе программного обеспечения, тем самым позволив снизить системные требования до уровня офисного компьютера. Данная возможность позволяет пользоваться системой управления патрульно-постовой службе любого города, без необходимости приобретать специальное оборудование в виде аппаратного сервера.

Выводы. Выполнен анализ работы патрульно-постовой службы милиции, произведен анализ существующих систем управления силами патрульно-постовых служб милиции, а также разработано программное обеспечение системы управления патрульно-постовой службой милиции. Применение результатов работы позволяет повысить эффективность использования сил и средств ППС при выполнении должностных и функциональных обязанностей за счет сокращения времени реагирования на происшествия и совершённые преступления.

Список литературы

1. Цветков В.Я. Геоинформационные системы и технологии. Серия “Диалог с компьютером”. – М.: Финансы и статистика, 1998. -286 с.
2. Кашенко Н.А., Попов Е.В., Чечина А.В. Геоинформационные системы. - Нижний Новгород ННГАСУ, 2012. -131с.
3. Мыльников Д. Геоинформационные платформы третья редакция 2012, -44с.
4. Сайт ГИС-Ассоциации, <http://gisa.ru/>
5. Геоинформационные системы, <http://www.dataplus.ru/>
6. Геоинформационные системы, <http://www.gisok.spb.ru/>



**СИСТЕМНЫЙ АНАЛИЗ
И УПРАВЛЕНИЕ
ОРГАНИЗАЦИОННЫМ
И ТЕХНИЧЕСКИМ
ОБЕСПЕЧЕНИЕМ**

СЕКЦИЯ 9

УДК 681.5

А.С. Бровкина, Ю.К. ОрловДонецкий национальный технический университет, г. Донецк
кафедра системного анализа и моделирования**СИНТЕЗ ИНФОРМАЦИОННО-СПРАВОЧНОЙ
СИСТЕМЫ ДИАГНОСТИРОВАНИЯ И ЛЕЧЕНИЯ
БОЛЕЗНЕЙ ДЛЯ МЕДИЦИНСКИХ ЗАВЕДЕНИЙ****Аннотация**

Бровкина А.С., Орлов Ю.К. Синтез информационно-справочной системы диагностирования и лечения болезней для медицинских заведений. Особенностью данной полезной модели является единый способ формализации всей имеющейся общей медицинской и специализированной клинической информации. Клиническая информационная система в автоматизированном он-лайн режиме функционирования предназначена для поддержки принятия врачебных решений при проведении дифференциальной диагностики и выборе комплексных корригирующих методов лечения.

Постановка задачи. В настоящее время информатизация охватывает всё более широкие сферы человеческой деятельности. При этом темпы роста цифровой информации значительно опережают темпы роста информации на бумажных носителях. Медицинская информация имеет свою специфику, и увеличение ее объемов сопровождается определенными проблемами, которые обосновывают необходимость создания медицинских информационных систем (МИС). Они отличаются от экономических или технических информационных систем, что создает дополнительные сложности при их разработке и внедрении. Отличие проявляется в специфике объектов информатизации, например, системы диагностического кодирования становятся в настоящее время более универсальными, но детальная номенклатура признаков и

симптомов, форматы для регистрации данных, а также организация записей определяются индивидуально; отсутствует стандартизация в терминологии, формате, шкалах измерения медицинских данных и т. д.

Цель статьи – провести анализ распространённых вирусных респираторных заболеваний и варианты их самостоятельного лечения. Основной целью создания системы является обеспечение эффективной информационной поддержки процесса управления системой медицинской помощи, а также процесса оказания медицинской помощи.

Постановка задачи исследования.

Задачи исследования:

1. Исследовать теоретические и прикладные аспекты системного анализа в построении информационных систем и СППР, состояние и уровень развития отечественных МИС и СППР.

2. Посредством анализа и синтеза процессов принятия управленческих решений формализовать принятие решений и построение СППР применительно к медицине.

3. Разработать методологические принципы использования современных технологий вероятностно-статистического моделирования в выявлении новых знаний. Предложить технологию решения задачи классификации объектов по группам однородности и применить для формализации слабоструктурированных медицинских проблем идентификации тяжести состояния больных.

4. На примере медицины показать состоятельность методологии анализа и синтеза систем принятия решений при создании эффективных СППР в различных сферах практической, исследовательской и образовательной деятельности человека, построив СППР для идентификации тяжести состояния больных.

5. Оценить эффективность разработанной СППР, применив ее для идентификации тяжести состояния больных, страдающих респираторными заболеваниями.

К сожалению, отечественная медицина и здравоохранение не в полной степени соответствуют требуемому уровню информатизации. Несмотря на указанные сложности, информатизация - объективный процесс, поэтому МИС все же медленно, но эволюционируют. Очень часто для принятия медицинских решений характерны недостаточность знаний, ограниченность временных ресурсов, отсутствие возможности привлечения компетентных экспертов, неполнота информации о состоянии больного. Указанные факторы являются причинами врачебных ошибок, которые могут привести к дальнейшей потере здоровья пациента. Поэтому наряду с разработкой МИС важной является задача создания медицинских систем поддержки принятия решений (СППР), которые являются информационными системами, функционирующими автономно либо в составе МИС.

Эта справочная система представляет собой процессорный узел, связанный с общей базой, используемой для подключения базы данных диагнозов, справочной системы из структурированной базы данных с подсистемами данных клинических характеристик симптомов, анатомических объектов, сведений о расположении анатомических объектов и клинических показателей выраженности, и системы ввода и регистрации результатов клинических исследований, при этом этот узел выполнен с функцией отбора из каждой подсистемы сведений, совпадающих со сведениями по результатам клинических исследований на основании использования критерия точности совпадения, критерия достоверности совпадения, критерия вероятности совпадения и критерия отрицания, и с функцией отбора из базы данных диагнозов сведений, соответствующих по крайней мере одному или нескольким диагнозам, клинические показатели которого совпадают с логически отобранными сведениями из структурированной базы данных. Поэтому данная работа, направленная на построение методологии создания СППР на основе современных технологий статистического моделирования, хранения и представления данных, является актуальной.

Каждый год острыми респираторными заболеваниями (ОРЗ) болеют от 20 до 40 миллионов человек, из которых 45–60% дети. Однако диагностика и лечение острых респираторных заболеваний нередко вызывают определенные затруднения у специалистов, поэтому эпидемии ОРЗ до сих пор остаются одной из наиболее значимых медицинских и социально-экономических проблем. Грипп и ОРВИ относятся к числу наиболее массовых инфекционных заболеваний, на их долю ежегодно приходится до 90-95% в структуре регистрируемой инфекционной заболеваемости. По данным статистических наблюдений каждый взрослый человек в среднем в год болеет в 2 раза респираторными инфекциями, школьник – 3 раза, ребенок дошкольного возраста – 6 раз.

Решение задач и результаты исследований. Создание Системы основывается на следующих принципах:

- однократный ввод и многократное использование первичной информации;
- использование электронных документов;
- обеспечение совместимости (интероперабельности) медицинских информационных систем;
- создание прикладных информационных систем по модели "программное обеспечение как услуга";
- обеспечение информационной безопасности и защиты персональных данных в соответствии с требованиями законодательства;

В рамках создания Системы обеспечивается автоматизация процессов формирования первичной медицинской документации в электронном виде с возможностью учета фактов оказания медицинской помощи.

Автоматизации подлежит возможность получения пользователями, а также организациями следующей информации:

- сведения о заболеваниях, методах их профилактики и лечения, о ведении здорового образа жизни, сведения о лекарственных средствах и изделиях медицинского назначения, новостную информацию;

- данные медицинской статистики, сведения о санитарно-эпидемиологической ситуации и действиях в случае ее ухудшения;
- сведения о лицензируемых видах деятельности в медицине и фармацевтике;
- сведения о медицинских организациях, кадровых и иных ресурсах в здравоохранении;

При внедрении информационных систем и СППР актуальной является проблема их функциональности и пригодности, которая выражается в том, что чем функциональнее система, тем она более сложна, а значит и менее пригодна для практического применения. Для снижения остроты проблемы необходимо подойти к разработке, внедрению информационных систем и СППР с позиций системного анализа, одним из этапов которого является математическое моделирование, включающее построение и анализ детерминированных и вероятностно-статистических моделей.

Особую актуальность для вероятностно-статистического моделирования приобретают методы многомерного статистического анализа, при помощи которых можно строить не только оптимальные планы сбора, систематизации и обработки данных, но и выявлять характер и структуру взаимосвязей между компонентами исследуемого многомерного признака.

Определенную помощь в принятии решения может оказать врачу компьютерное моделирование состояния больного в зависимости от выбора той или иной тактики или стратегии лечения.

Существующие системы управления базами данных (СУБД) позволяют хранить, обрабатывать и предоставлять пользователю информацию в удобном для него виде. Применительно к медицине в СУБД может храниться вся информация о больном в виде электронной истории болезни (ЭИБ).

На основе принципов системного анализа предложена методология построения эффективных СППР, которая может быть использована для создания СППР в различных областях практической деятельности человека.

Выводы. Проведен анализ методов организации справочной системы и диагностики и лечения заболеваний. Данная система относится к медицине и может быть использована в поликлинической и стационарной медицинской помощи в качестве системы дифференциации и анализа зарегистрированных клинических данных в том числе составе автоматизированной клинической информационно-интеллектуальной справочной системы - АКИИСС, либо иных информационных систем, позволяющей проводить дифференциацию клинических и инструментально-лабораторных данных для дифференциации предварительных и окончательных диагнозов.

Список литературы

1. Халафян А.А. STATISTICA 6. Статистический анализ данных: Учебник, - М.: Бином, 2007. 512 с.
2. Халафян А.А. Современные статистические методы медицинских исследований. Монография. - М.: ЖИ (URSS). 2008. 320 с.
3. Халафян А.А. STATISTICA 6. Статистический анализ данных: Учебник. Второе издание. - М.: Бином, 2009. 528 с.
4. Сштиков В.С., Халафян А.А. Системный анализ и современные информационные технологии в медицинских системах поддержки принятия решений. Монография, - М.: Бином, 2009, 362 с.
5. Халафян А.А. STATISTICA 6. Математическая статистика с элементами теории вероятностей: Учебник. - М.: Бином, 2010. 4

УДК 65.011.56

Т.В. Ванжа, Ю.К. Орлов

Донецкий национальный технический университет
кафедра системного анализа и моделирования

Email: sam_dntu@i.ua

РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ ГОРНЫМ ПРЕДПРИЯТИЕМ В УСЛОВИЯХ РИСКОВ

Аннотация

Ванжа Т.В., Орлов Ю.К. Разработка системы управления горным предприятием в условиях рисков. В статье рассматривается возможность применения функций риск-менеджера для организации управления риском, понятие и его сущность. Рассматриваются различные методы оценки и минимизации производственных и профессиональных рисков на горном предприятии.

Ключевые слова: сущность, система управления риском, риск-менеджер, структурная схема, вероятностно-статистические методы.

Постановка проблемы В последние годы роль и значение управления рисками как инструмента снижения потерь и повышения эффективности национальных экономик во всем мире постоянно возрастает. Значение этого инструмента возрастает, прежде всего, из-за роста самих рисков, что является общемировой тенденцией, обусловленной усложнением всех сфер функционирования современного общества.

Горное предприятие является объектом повышенной опасности. При этом безопасная работа предприятия во многом определяется управлением риска на нём. Любая авария обусловлена несколькими факторами: внешним воздействием, ошибками персонала, отказами оборудования, как производственного, так и противоаварийного.

Цель: оценка и минимизации рисков на горном предприятии при помощи вероятностно-статистических методов оценки.

На рисунке 1 изображена схема риска. Помимо самого человека на его деятельность влияют факторы I-VI, создающие в ней ситуации неопределенности и риска. Действия и решения человека могут повлиять как в лучшую сторону на результаты деятельности, так и в худшую, разрядить ситуацию неопределенности, изменить цель проекта. Риск существует как на этапе выбора действий (принятия решений), так и на этапе реализации проекта.

Организация управления риском на горном предприятии

Управление рисками на горном предприятии включает в себя четыре основных процесса: идентификация, анализ, планирование и контроль рисков. Функционирование системы представлено на рисунке 2.



Рисунок 2 – Функционирование системы управления рисками на предприятии

Эффективное управление риском требует не только внимательного наблюдения за размером риска, но также стратегию минимизации убытков. Для борьбы с рисками применяются известные методы управления [2].

Целью снижения рисков является уменьшения неопределенности во внешней среде и внутренней структуре предприятия, повышение точности прогноза количественных значений факторов риска. Для этого, с одной стороны, необходимо повышать точность прогнозов факторов риска путем совершенствования информационного обеспечения управления рисками (предприятием), а с другой - применять известные экономические, организационные методы снижения рисков представленных на рисунке 3.

Для управления риском на горных предприятиях создается специальное подразделение – отдел управления риском (ОУР), возглавляемое риск-менеджером, т.е. руководителем, который занимается исключительно проблемами управления риском и координирует действия всех подразделений в плане регулирования риска и обеспечения компенсации возможных потерь и убытков.



Рисунок 3 – Методы управления риском

Можно выделить три основных организационных аспекта структуры управления риском[3]:

- деятельность ведущего риск-менеджера;

- деятельность подразделения (отдела) управления риском;
- взаимосвязь подразделения с другими структурами организации.

На рисунке 4 показан пример алгоритма управления риском в системе управления горным предприятием.

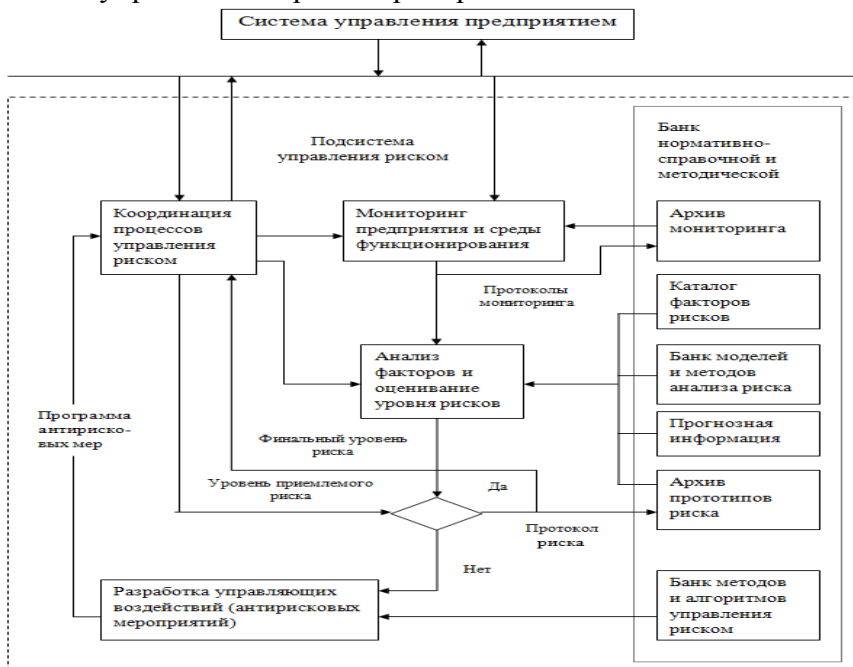


Рисунок 4 – Структурная схема алгоритма управления риском на предприятии

Для построения модели «черный ящик» необходимо определить границы системы управления, внешнюю среду, входы и выходы системы.

Граница системы – само горное предприятие.

В качестве внешней среды для системы управления риском выступает среда функционирования предприятия.

Входами в систему есть внешние факторы, которые несут за собой ситуации риска (политические, социально-экономические, экологические, научно-технические).

К выходам системы относятся программы антирисковых мер, т. е. регулирование риска и обеспечение компенсации возможных потерь и убытков. Модель «Чорний ящик» представлена на рисунке 5.

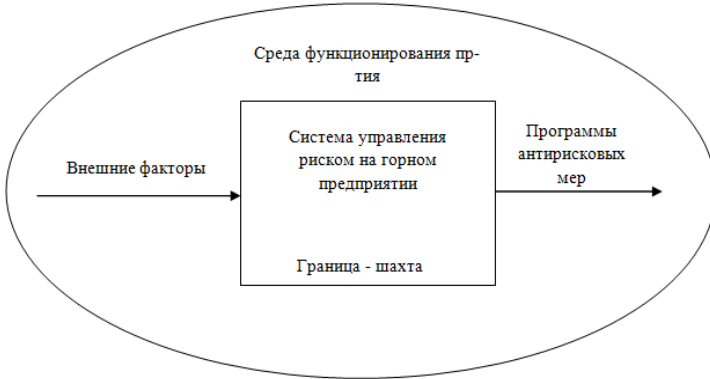


Рисунок 5 – Модель «чорний ящик» системы управления рисками
 Функциональная модель системы управления рисками
 представлена на рисунке 6.

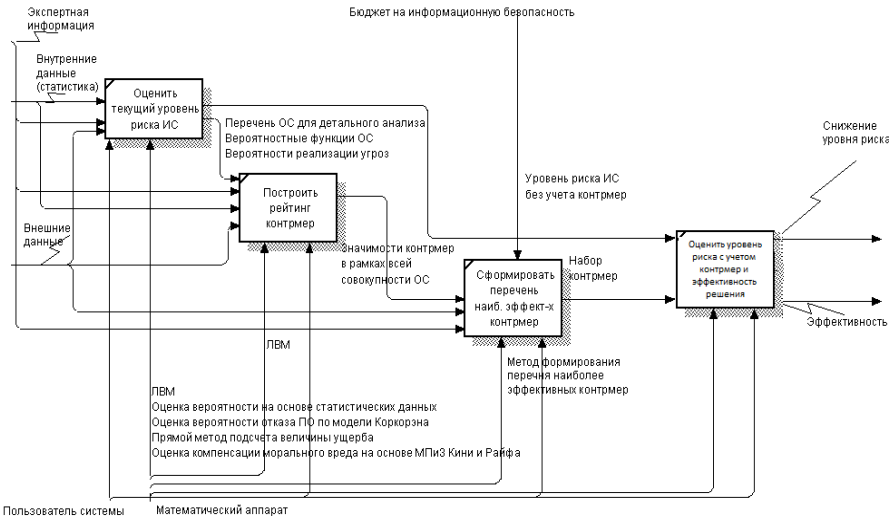


Рисунок 6 – Функциональная модель системы

Вероятностно-статистические методы оценки производственных рисков На горном предприятии профессиональный риск можно оценивать методами теории вероятностей и математической статистики[4]. Статистический метод обеспечивает наибольшую точность оценки (при наличии достаточной статистики). В соответствии со статистическим методом показатель профессионального риска получения работником травмы в результате несчастного случая на производстве можно оценить по формуле:

$$v_{\text{нс}} = \frac{n_{\text{нс}}}{N_p}, \frac{1}{\text{чел.год}} \quad (2)$$

где $n_{\text{нс}}$ – число несчастных случаев на производстве в год на предприятии (в отрасли, на территории);

N_p – численность персонала, подвергающегося рассматриваемому риску в оцениваемом году.

Статистический метод по объединенной выборке применяется для обеспечения требуемой точности оценки показателя профессионального риска получения травмы. При недостаточности фактически выявленных событий в течение одного года можно увеличивать объем наблюдений за счет увеличения интервала наблюдения, т.е. путем объединения имеющихся статистик за ряд последовательных лет[5].

Поскольку наступление неблагоприятного события (инцидента, несчастного случая, гибели или обобщающего их условного события) на рабочем месте работника является редким событием, то оценка частоты события для i -го рабочего места определяется по формуле:

$$\lambda_i = \frac{n_i}{t_i}, 1/\text{лет} \quad (3)$$

где n_i – число событий на i -м рабочем месте за время t_i , лет.

Усредненная частота несчастных случаев со смертельным исходом для совокупности из N_p работающих в одинаковых условиях оценивается по формуле:

$$\lambda = \frac{n}{S}, \quad (4)$$

где $n = \sum_{i=1}^{N_p} n_i$ – суммарное число событий;

$S = \sum_{i=1}^{N_p} t_i$ – суммарная наработка всех N_p работающих.

Суммарную наработку можно представить в следующем виде:

$$S = N_p \cdot T, \quad (5)$$

где T – средняя наработка на одного работающего за один год.

Вероятностно-статистический метод применяется при отсутствии достаточной статистики и основан на привлечении математических моделей [5]. Полная индивидуальная вероятность пострадать от несчастного случая вычисляется по формуле:

$$v_{\Sigma} = \frac{n_{\Sigma}}{N_p}, \quad (6)$$

где n_{Σ} – общее число несчастных случаев на производстве на рассматриваемом предприятии в оцениваемом году;

N_p – численность персонала, подвергающегося рассматриваемому риску в оцениваемом году.

Показатель профессионального риска смерти в соответствии с вероятностно-статистическим методом (даже при отсутствии несчастных случаев со смертельным исходом на данном предприятии) может быть оценен по формуле:

$$v_{\text{см}} = v_{\Sigma} \cdot q, \quad (7)$$

где q – доля несчастных случаев со смертельным исходом от общего числа несчастных случаев.

Соотношения между значениями весовых коэффициентов подбираются эмпирически, исходя из реального уровня производственной безопасности на предприятии и целей анализа риска.

Выводы Управление производственными рисками это комплекс организационных и технических мероприятий, которые

в соответствии с принципами современных систем менеджмента должны основываться не на прошлом опыте, интуиции или гениальных догадках, а на результатах количественного анализа. Управление рисками в обязательном порядке подразумевает оценивание (измерение) рисков до применения и после защитных мер. Очень важно, чтобы измерение рисков в обоих случаях проводилось по одной и той же методике, по возможности, сводящей к минимуму влияние на результат оценки субъективных факторов.

В настоящей статье представлены возможные подходы к оцениванию профессиональных рисков на уровне отрасли горнодобывающей промышленности. Несмотря на то, что простейшие методики оценки рисков на рабочих местах широко применяются в современных системах менеджмента производственной безопасности и здоровья, последним двум требованиям они не удовлетворяют. Поэтому задача разработки научно обоснованной простой методики оценки рисков на сегодняшний день является весьма актуальной.

Список литературы

1. Лускатова О.В. Оценка риска и экономической устойчивости горных предприятий на основе нечеткой логики / Норильский индустр. ин-т. - Норильск, 2004. - 221 с.
2. Мамаева, Л.Н. Управление рисками: Учебное пособие / Л.Н. Мамаева. - М.: Дашков и К, 2013. - 256 с.
3. Хенли Э. Дж., Кумamoto X. Надёжность технических систем и оценка риска. М.: Машиностроение, 1984. –528 с.
4. Профессиональный риск для здоровья работников (Руководство) / Под ред. Н.Ф. Измерова и Э.И. Денисова. М.: Тровант, 2003. 448 с.
5. Радаев Н.Н. Повышение точности прогноза вероятности катастроф за счет учета неоднородных статистических данных по ущербу// Автоматика и телемеханика, 2000 г., №3, с. 183-18

УДК 512.07

К.В.Радевич, Ю.К.Орлов.

Донецький національний технічний університет, м. Донецьк
кафедра системного аналізу і моделювання

РОЗРОБКА СИСТЕМИ МЕРЕЖЕВОГО ПЛАНУВАННЯ ВИРОБНИЦТВА ПРИ БУДІВНИЦТВІ АВТОДОРІГ І МОСТІВ

Анотація

Радевич К.В. Орлов Ю.К. Розробка системи мережевого планування при будівництві автодоріг і мостів. Мета науково-дослідної роботи полягає в розробці моделі мережевого планування при будівництві автодоріг і мостів. Для вирішення проблеми недостатнього контролю з боку ГУБіКО служб, які займаються будівництвом подібних об'єктів запропоновано виконати наступне:

- 1) провести системний аналіз підприємства, для більш детального розгляду проблеми;*
- 2) проаналізувати методи мережевого планування для пошуку найкращого методу;*
- 3) вирішити проблему із застосуванням обраного метода.*

***Ключеві слова:** системний аналіз, планування, математична модель, синтез структури і алгоритму системи управління.*

Постановка проблеми. Проаналізувати роботу ГУБіКО, зробити системний аналіз підприємства. Вирішити проблему недостатнього контролю над виконавцями при будівництві автодоріг і мостів, для чого скласти мережевий графік і розрахувати грани і пізній час початку робіт.

Мета статті – створення і розробка системи мережевого планування виробництва при будівництві автодоріг і мостів, в якому витрати і час мінімізовані, а якість – максимізована.

Постановка задачі та дослідження. Оптимізувати роботу ГУБіКО шляхом вирішення проблеми недостатнього контролю при будівництві автодоріг і мостів. Для існуючої моделі системи скласти мережевий графік із зазначенням раннього і пізнього часу завершення роботи.

Вирішення задачі та результати досліджень. Головним недоліком ГУБіКО є те що контроль за будівництвом не здійснюється автоматично. Співробітники змушені підраховувати коли саме треба перевіряти об'єкт. Тому у дипломній роботі пропонується виконувати контроль за проектом автоматизованим шляхом. Такий підхід однозначно скоротив би час і гроші ГУБіКО. Використання програми Microsoft Project дозволить швидко порахувати, який час необхідно для будівництва (ремонту) моста, заробітну плату робітників і багато іншого.

Нижче наведені допоміжні таблиці для виконання проекту в програмі Microsoft Project. Метод мережевого планування, використовуваний при складанні таблиць і графіків, називається методом критичного шляху. Його алгоритм описаний вище. У таблиці 1 визначимо черговість виконання робіт, їх передбачувану тривалість, а також попередника.

Таблиця 1 – Вихідні дані для структурного планування

	Назва роботи	Тр-сть	Поп-ник
1	Початок проекту	0	Немає
	Етап 1: Попередня підготовка, збір первинної інформації		
2	Отримання конкурсної документації та її розгляд	20	1
3	Обстеження об'єкта та збір даних	14	2
4	Ухвалення рішення про можливість і доцільність будівництва силами підприємства	10	1
5	Участь у тендерних торгах (подача заявки на участь; надання всіх необхідних документів)	30	3,4
6	Підписання договору на будівництво (капітальний ремонт) за умови перемоги у тендерних торгах	5	5

	Етап 2: Розробка та узгодження проєктів виконання робіт		
7	Отримання затверджених проєктів на будівництво	10	5
	Розробка проєктів виробництва робіт		
8	Розробка календарного плану виробництва робіт	5	7
9	Розробка графіка постачання обладнання та матеріалів	2	8
10	Розробка графіка руху робочої сили	2	7
11	Розробка технологічних карт (схем)	10	10
12	Ухвалення рішення щодо забезпечення техніки безпеки	5	6
13	Узгодження проєктів виконання робіт з усіма зацікавленими службами та організаціями	25	6
	Етап 3: Будівництво		
	Підготовчі роботи:		
14	Підготовка будівельного майданчика	5	12,13
15	Електропостачання та водопостачання буд майданчика	10	14
16	Установка побутових приміщень	10	9,11
17	Устрій майданчиків і складів для зберігання будівельних матеріалів і конструкцій;	25	15,16
18	Винесення або перебудова комунікацій, які знаходяться на території будівництва	30	17
19	Устрій з'їздів, робочих майданчиків і робочих містків.	15	17
	Будівельно-монтажні роботи:		
20	Устрій фундаментів;	25	19
21	Спорудження опор;	50	20
22	Монтаж прогонових будов;	65	21
23	Устрій мостового полотна;	60	19
24	Устрій перильних огорож, сходових сходів;	25	23
25	Зміцнення укосів насипу;	15	18
26	Рекультивация території.	10	25
	Етап 4. Здача об'єкта в експлуатацію.		
27	Підготовка об'єкта до введення в експлуатацію;	5	26
28	Передача виконавчої документації замовнику;	1	24,27
29	Здача об'єкта в експлуатацію (комісійна підписання акта введення в експлуатацію)	2	22,23,28
30	Кінець проєкту	0	29

За допомогою таблиці 1 порахуємо ранній час початку робіт (рисунок 1).

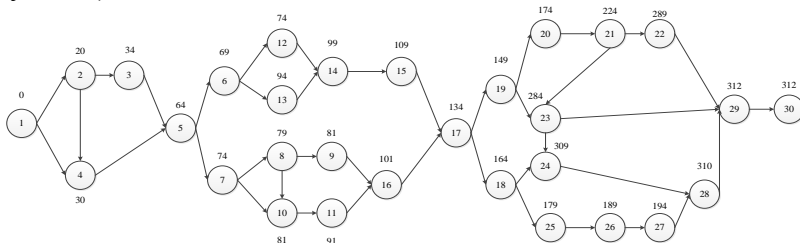


Рисунок 1 – Ранній час початку роботи

За допомогою таблиці 1 порахуємо пізній час початку робіт (рисунок 2). Критичний шлях на малюнку позначений пунктирною лінією.

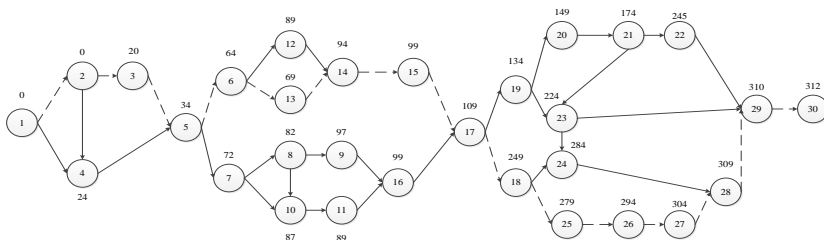


Рисунок 2 – Пізній час початку роботи

За допомогою програми Microsoft Project можна зробити мережевий графік та виконати планування всіх аспектів проекту.

За допомогою програми також можна роздрукувати звіти, в яких є всі необхідні відомості про проект.

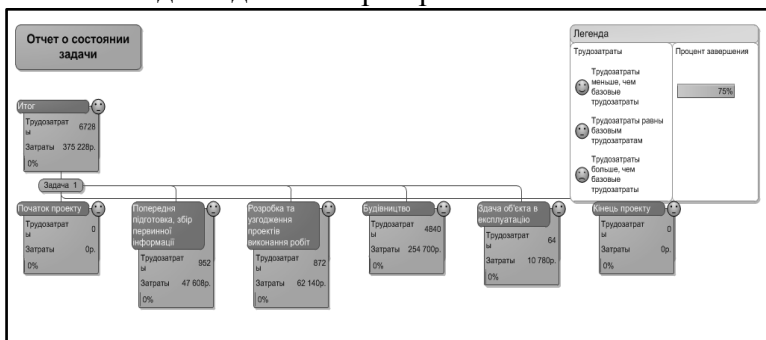


Рисунок 3– Звіт про стан завдання

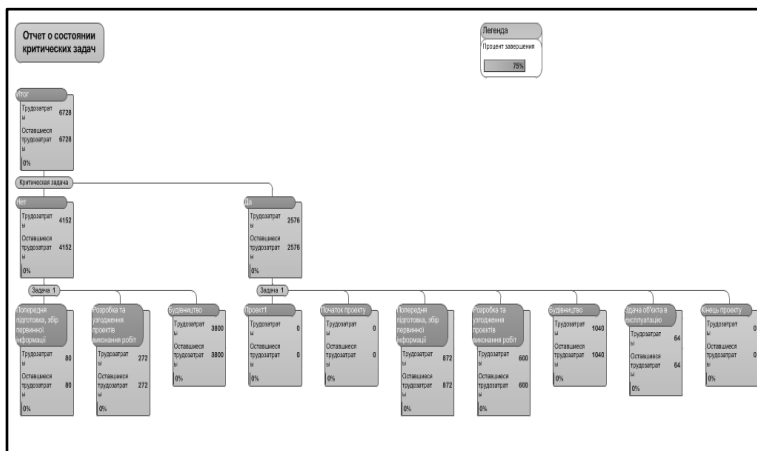


Рисунок 4 – Звіт про стан критичних задач

Мережевий графік з критичними задачами зображено на
рисунку 5.

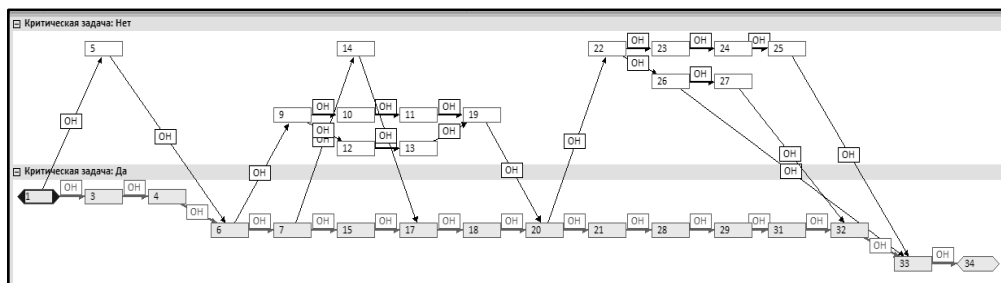


Рисунок 5 – Мережевий графік

Висновки.

Навіть не дивлячись на істотний недолік програми Microsoft Project, громіздкість таблиць і графіків, її рекомендовано використовувати при плануванні будівництва автодоріг і мостів. Програма дозволить зробити всі необхідні розрахунки, а також переглянути потрібну інформацію за конкретний період часу. Використання Microsoft Project істотно скоротить час роботи над подібними проектами співробітників ГУБіКО.

Перелік посилань

1. Лучко Й.Й.Будова та експлуатація штучних споруд / Й.Й.Лучко, О.С.Распопов. – Львів: Каменяр, 2010. – 890 с.
2. Експлуатація і реконструкція мостів / [Страхова Н.Є., Голубев В.О., Ковальов В.В., Тодіріка В.В.]. – Транспортна академія України, 2002. –403 С.
3. Колесник Д. Ю., Шкуратовський А. О., Парубець М. Г., Коваль П. М. Економічна доцільність вторинного захисту залізобетонних конструкцій мостів. Автошляховик України. – 2005. – №4. – С. 34–39.
4. Фаль А. Є. Трещиностійкість та експлуатаційні якості залізобетонних плит проїзної частини автодорожніх мостів / А. Є.Фаль. – Львів, 2004. – 190 С.
5. Крамер Е. Л. Итоги науки и техники. Автомобильные дороги / Е.Л Крамер. – М.: ВИНТИ, Том 9. – 1990. – С. 64–171.
6. Голенко Д. И. Статистические методы сетевого планирования и управления/ Д.И.Голенко. – Москва: издательство Наука, 2000. – 400 С.
7. Кофман А. Сетевые методы планирования и их применение /А. Кофман, Г. Дебазей. –Москва: Издательство Прогресс, 2000. –180 С.
8. Кульгин Н.Б. Инструменты управления проектами: ProjectExpert и MicrosoftProject / Н.Б. Кульгин – СПб.: БХВ–Петербург, 2009. – 160 С.
9. Головне управління благоустрою та комунального обслуговування [Електронний ресурс]. – Режим доступу: <http://blago.dn.ua/>